SPEECH TO CHART: SPEECH RECOGNITION AND NATURAL LANGUAGE PROCESSING FOR DENTAL CHARTING

by

Regina (Jeannie) Yuhaniak Irwin

Bachelor of Arts, Pennsylvania State University, 2000

Master of Science in Biomedical Informatics, University of Pittsburgh, 2006

Submitted to the Graduate Faculty of

School of Medicine in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2009

UNIVERSITY OF PITTSBURGH

SCHOOL OF MEDICINE

This dissertation was presented

by

Regina (Jeannie) Yuhaniak Irwin

It was defended on

September 28, 2009

and approved by

Dr. Titus Schleyer, Associate Professor, Center for Dental Informatics

Dr. Heiko Spallek, Assistant Professor, Center for Dental Informatics

Dr. Brian Chapman, Assistant Professor, Department of Biomedical Informatics

Dr. Peter Haug, Professor, Department of Biomedical Informatics, The University of Utah

Dissertation Advisor: Dr. Wendy Webber Chapman, Assistant Professor, Department of Biomedical Informatics

SPEECH TO CHART: SPEECH RECOGNITION AND NATURAL LANGUAGE PROCESSING FOR DENTAL CHARTING

Regina (Jeannie) Yuhaniak Irwin, PhD

University of Pittsburgh, 2009

Typically, when using practice management systems (PMS), dentists perform data entry by utilizing an assistant as a transcriptionist. This prevents dentists from interacting directly with the PMSs. Speech recognition interfaces can provide the solution to this problem. Existing speech interfaces of PMSs are cumbersome and poorly designed. In dentistry, there is a desire and need for a usable natural language interface for clinical data entry.

Objectives. (1) evaluate the efficiency, effectiveness, and user satisfaction of the speech interfaces of four dental PMSs, (2) develop and evaluate a speech-to-chart prototype for charting naturally spoken dental exams.

Methods. We evaluated the speech interfaces of four leading PMSs. We manually reviewed the capabilities of each system and then had 18 dental students chart 18 findings via speech in each of the systems. We measured time, errors, and user satisfaction. Next, we developed and evaluated a speech-to-chart prototype which contained the following components: speech recognizer; post-processor for error correction; NLP application (ONYX) and; graphical chart generator. We evaluated the accuracy of the speech recognizer and the post-processor. We then performed a summative evaluation on the entire system. Our prototype charted 12 hard tissue exams. We compared the charted exams to reference standard exams charted by two dentists.

Results. Of the four systems, only two allowed both hard tissue and periodontal charting via speech. All interfaces required using specific commands directly comparable to using a mouse. The average time to chart the nine hard tissue findings was 2:48 and the nine periodontal findings was 2:06. There was an average of 7.5 errors per exam. We created a speech-to-chart prototype that supports natural dictation with no structured commands. On manually transcribed exams, the system performed with an average 80% accuracy. The average time to chart a single hard tissue finding with the prototype was 7.3 seconds. An improved discourse processor will greatly enhance the prototype's accuracy.

Conclusions. The speech interfaces of existing PMSs are cumbersome, require using specific speech commands, and make several errors per exam. We successfully created a speech-to-chart prototype that charts hard tissue findings from naturally spoken dental exams.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

I would like to express my deep and sincere gratitude to my committee chair, Wendy W. Chapman, Ph.D. She accepted me mid-stream in my research and supported all of my ideas. Dr. Chapman continually motivated and challenged me. Her knowledge, skills, understanding, encouragement, and personal guidance have been vital to my growth as a person and essential for the completion of this thesis.

I wish to express my warm and sincere thanks to Titus Schleyer, D.D.S., Ph.D. and Heiko Spallek, D.D.S., Ph.D. I began my Ph.D. training knowing nothing about dentistry. Without their guidance, ideas, and abundance of dental knowledge this dissertation would not exist. Both Dr. Schleyer and Dr. Spallek fostered my growth. Many of my achievements and acquired skills exist only because they encouraged and supported my work both within and outside the scope of my research.

I am deeply grateful to Brian Chapman Ph.D. Dr. Chapman's advice and support was essential in the completion of this thesis. He was always available, especially during the difficult times when the writing of this thesis became tedious. He provided brilliant programming ideas, personal guidance, and a multitude of research suggestions. Dr. Chapman has had a major influence on my career in the field of Biomedical Informatics and my growth as a person.

I warmly thank Peter Haug M.D., for his valuable advice and outstanding ideas. His discussions about my work have guided many aspects of this thesis. Without Dr. Haug's insight, there probably would be no speech-to-chart prototype and my work would feel incomplete.

I am deeply grateful to Lee Christensen and Henk Harkema, Ph.D. Both Mr. Christensen and Dr. Harkema were essential in my work. Without their help, we would not have ONYX, the main element of our speech-to-chart prototype. They were both always available whenever I needed more annotations, to discuss an issue, or a bug fixed. Their skills and insight have provided a significant addition to this dissertation.

During this work I have collaborated with many colleagues from the Department of Biomedical Informatics (DBMI) for whom I wish to extend my warmest thanks. To Heather Piwowar, Holly Berty, and Thankam Thyvalikakath, thank you for all of the great talks, help with revision of drafts, research suggestions and personal support. Everyone should have friends like you for support and guidance; I feel truly lucky to have you in my life. To Toni Porterfield, thank you for all of your help and always being there to answer my questions. To all of my DBMI colleagues, thank you making our department and this research facilities one of the best places to work.

I owe my loving thanks and so much more to my husband Jeff Irwin. Without his encouragement, support, and understanding it would have been impossible for me to finish this work. My special gratitude is due to my mom, Sherry Yuhaniak, my brother, Andy Yuhaniak, his wife, and Jeff's family who have become my family, for their loving support. They listened intently when I discussed my research, the hard times, and the wonderful experiences of being a grad student. Their love is essential in my life.

Finally, the financial support from the NIDCR, the NLM, and DBMI is gratefully acknowledged. Without this funding I would not have completed my studies.

1.0 INTRODUCTION

1.1 DESCRIPTION OF THE PROBLEM

During care, dental clinicians are restricted in their use of a keyboard and a mouse, primarily because of infection control concerns but also because they use their hands for procedures and exams. Moreover, the office space and setup make it difficult to have a keyboard in close proximity. Therefore, they often use auxiliary personnel to record data in the patient chart. A solution to this problem that is being employed in medicine [1-6], is using speech recognition applications to interact with the clinical computer. If dental clinicians can immediately access and enter data while chairside, they can realize the benefits of improved documentation, increased efficiency, automatically captured billing information, and chairside decision support, diagnosis, treatment planning, and scheduling.

A survey of U.S. general dentists on clinical computer use singled out speech recognition for facilitating direct charting as one of the most desirable improvements in existing applications [7]. The study also showed that 13 percent of all dental offices surveyed had used speech input; however, many tried and discontinued using the technology. Those who discontinued using speech did so because of technical problems with speech recognition (57%), lower efficiency compared to other data entry methods (13%), usability problems (9%), and other issues (22%)

[7]. These numbers indicate that there may be significant barriers to using the speech modules of existing dental systems.

The speech recognition functionality of existing dental software typically implements command-and-control functionality as well as the transcription of free text. The command-and-control functionality supports two types of activities: (1) navigating within the application—for example, to select a specific patient—and (2) entering structured data in text fields, list boxes, radio buttons and checkboxes. Transcription is used primarily for dictating progress notes, surgical reports and similar textual information. To enter data in a hard tissue or periodontal chart, the dentist must speak a multitude of very specific commands, and the interaction is directly comparable to using a mouse. Conversely, dentists generally dictate findings in a fairly unstructured interaction with auxiliary personnel—for example, a dentist might dictate a cavity as, "Tooth number 3 has caries on the mesial." In a leading charting system to chart a mesial caries on tooth 3 he must say: "conditions," "move down 9," "move down 8," (for moving to the caries item on a list of conditions) "OK," "mesial," "OK," "existing." (Example produced using: Dentrix Chart Version 10.0.36.0, Dentrix Voice Version 4.0, Dentrix Dictation Version 2.0, all Henry Schein, Melville, NY). The crucial flaw in the design of most existing speech applications is that the applications do not communicate in the natural language pattern of the user.

Because of the poor user interface and speech application design of current systems [8], most clinicians perform data entry by utilizing an assistant/hygienist as a transcriptionist. This situation is problematic in two respects. First, using auxiliary personnel prevents the dentist from interacting directly with the computer during clinical care, which reduces the potential benefit of clinical decision support systems. Clinical decision support systems are most effective when used directly by the decision-maker [9]. Second, using auxiliary personnel for data entry reduces

the efficiency of the dental office, because dental hygienists and assistants cannot perform value-added clinical task when they are engaged as computer operators. The absence of a flexible, robust, and accurate natural language interface is a significant barrier to the direct use of computer-based patient records by dental clinicians.

The long-term goal of our research is to develop a natural language interface that will allow clinicians to speak naturally as a means of entering data in a computer-based patient record without using the keyboard and mouse and without relying on an auxiliary. Before developing a natural language system, we must critically evaluate currently existing speech input systems in dentistry. The objectives of my research are:

1. evaluate the efficiency, effectiveness, and user satisfaction of the speech interfaces of four existing dental practice management systems and
2. develop and evaluate a speech-to-chart prototype for charting naturally-spoken dental exams.

A system that is designed to accommodate natural dictations will allow dentists to realize the benefits of interacting directly with the electronic patient chart at the point of care.

1.2 SIGNIFICANCE OF THIS RESEARCH

This work is highly significant to the field of dentistry. For the first time to our knowledge, researchers will explore the use of natural language as a method to document patient care in dentistry. Our preliminary research has shown that there is a demand for improved speech recognition at chairside [7] and that existing systems perform poorly [8]. The tools developed in this dissertation will enable the dental clinician to chart dental conditions at chairside while

eliminating infection control concerns, the need for structured input, and the need for a transcriptionist.

Once recognition accuracy reaches an acceptable level, several benefits of speech applications in dentistry would be realized [10]. First, data are immediately entered into the computer, saving the step of entering handwritten notes or transcribing an audio recording at a later time. Second, the potential for data entry errors is reduced because data are immediately validated by the computer and the data entry person via visual confirmation. Third, auxiliary personnel, who often function as transcriptionists, are freed up for other tasks. Our natural language processing-based dental charting application can lead to improved documentation, increased office efficiency, and a structured chart that will support chairside decision support, which all potentially lead to better patient outcomes.

This work is also significant to the field of biomedical informatics. Speech applications in medicine have primarily been developed for transcription, not for natural language processing-based data entry. To our knowledge, this project represents the only attempt to assess the combination of speech recognition and natural language processing-based (NLP) for clinical data entry at the point of care. The results of this research are therefore applicable to other medical disciplines with similar constraints and workflows as dentistry, such as otolaryngology and ophthalmology. Moreover, the NLP application we developed has potential for implementation at the point-of-care, which is rare for NLP applications in clinical medicine.

2.0 BACKGROUND

This section reviews the current state of dental chairside computing, the basics of speech recognition and speech recognition in medicine and dentistry. It continues with the basics of natural language processing and natural language processing in medicine. Next, we included the previously published work (with permission) of our development and evaluation of semantic representations for natural language processing. The semantic representations described are an integral component in the natural language application we developed for this project, called ONYX. Finally, this section ends with a discussion of ONYX.

2.1 COMPUTING IN DENTISTRY

As of 2006, 92.6 percent of all U.S. dentists were using a computer in their offices [11], but a recent study shows that only 25 percent of all general dentists use a computer in the clinical environment [7]. As of 2006, only 1.8 percent of all general dental practices in the U.S. maintained completely computer-based patient records [7]. Practice Management Systems, which are the dental applications that support computer-based patient records, allow dentists to perform electronic scheduling, treatment planning, patient education, charting, and storage of exam data, digital x-rays, patient histories and more [7]. A survey of U.S. general dentists on clinical computer use shows that the top three reasons for adoption of chairside computers are to

(1) improve data management—such as, direct entry of treatment plans and appointments (2) digital imaging—primarily digital radiology—and (3) improved efficiency—for example, scheduling directly in the operatory [7]. Participants also listed many barriers to and disadvantages of computers in the clinical environment—such as, insufficient operational reliability (e.g. system crashes), functional limitations of the software, the learning curve, cost, infection control issues, and insufficient usability. Further, when Practice Management Systems (PMS) were compared to paper charts, PMSs were found to have limited information coverage of clinical information and disassociated data fields [12], both inhibiting chairside computer use.

2.2 SPEECH RECOGNITION

Speech recognition is the process of converting spoken words into machine understandable input for further processing and/or transcription [13]. Speech recognition has been used for tasks like telephone voice-response systems [13]—which use small vocabularies and can handle multiple speakers—and for the much different task of transcribing continuous speech from one user with large-vocabulary recognition [14]. Transcription of continuous speech (in the form of a dictation) is what is needed for the charting of naturally spoken clinical exams. This section describes the processes of a simple speech recognizer as can be seen in Figure 1.

Continuous speech recognition systems use a microphone to capture a person's voice as series of acoustic waves (Item 1 in Figure 1) [15]. Those waves are then converted into a spectral representation that provides a sequence of vectors (Item 2 in Figure 1). Each vector is a representation of features for a time-slice of the voice signal (usually ten msecs) [14, 15]. An acoustic vector is used to identify the differences in the sounds of the voice signal. For example,

as seen in Figure 1, Item 2, the vector for the end of the sound /d/ contains very different features than the vector for the start of the sound /p/.

The next steps in a simple speech recognizer involve converting the speech sounds to text. A speech recognition engine typically relies on three knowledge sources: (1) an acoustic model that specifies the probability of a phoneme (speech sound) based on user training; (2) a dictionary that specifies the pronunciation of a particular word using one or more phonetic transcriptions; and (3) a grammar, or language model, that models the probability of a word given a history of previous words [14].

The first step is acquiring the phonemes (or phones) from the characteristics of the acoustic vectors. Based on a user training (i.e. the user reads a passage the system knows and the system learns how the user pronounces words) in the form of an acoustic model, the speech recognizer uses hidden Markov models (HMM) to process the vectors and identify phones [14]. A pronunciation dictionary—a collection of words and the phones that make up how they are commonly pronounced—then provides input to the HMMs to convert the phones into words. As the system processes the sound vectors, it scores possible phones and as more phones are added the system creates a score for possible words. For example, as seen in Figure 1, Item 3 based on similar sounds, the system could score the phoneme /p/ and the phoneme /t/ similarly thus creating the need to choose between the words “potato” and “tomato”. However, if the phoneme /p/ has a slightly higher score, the system should correctly transcribe the word as “potato”.

Finally, the system can use n-grams and language models to estimate the probability of a word given preceding words [14]. As seen in Figure 1, Item 4, the system can assign a higher probability to the word “potato” when it is preceded by the word “baked”. Thus, allowing the system to choose “potato” instead of “tomato”.

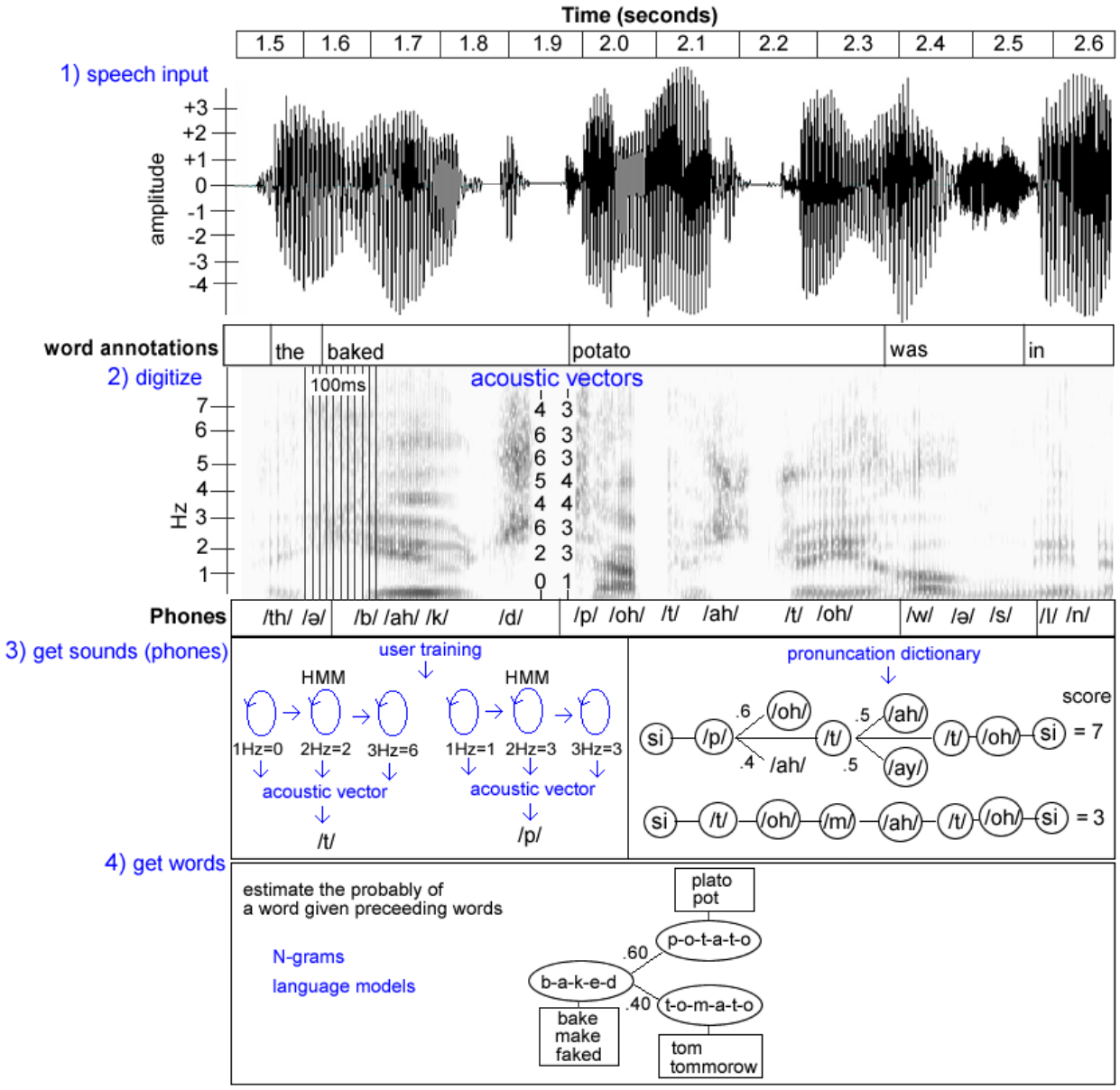

Figure 1. Process of a simple speech recognizer. Figure is modeled after Figure 7.2 in [15].

Speech recognition technology has made significant improvements over the years [15]. These improvements have encouraged those in the medical profession to adopt speech recognition as a means to automatically transcribe dictations. Some medical speech software providers claim recognition accuracy rates as high as 99 percent [16]. As speech recognition is tailored for domains like radiology and dentistry it can offer reduced report turnaround times, reduced staffing needs, and the ability to interact hands-free with charting systems [16].

2.3 SPEECH RECOGNITION IN MEDICINE

Speech recognition in medicine is used almost exclusively for transcription—that is, the act of converting spoken words to text. Systems in areas such as radiology, cardiology, endoscopy, and general practice can create instantly available computerized reports from naturally spoken dictations [1-6]. Results from a survey of 31 authors of papers on medical speech recognition indicate that health care organizations are optimistic about these applications and predict a trend towards increased adoption [3]. One healthcare system noted that 25 percent of their physicians use speech recognition for some data entry [17]. Most data entry with speech is not real-time entry.

With specialized vocabularies and clinician training, medical speech recognition applications are known to have up to a 95 percent accuracy rate [3]. At this accuracy level, implementing speech recognition applications can significantly reduce costs for human transcription and decrease turn-around time of report availability [3, 5, 6, 18-20]. Major barriers for the adoption of these systems include integrating systems into clinical workflow, clinicians having some difficultly training the systems (especially non-native English speakers), and the time needed to edit reports for errors [4, 19-22]. Overall, the benefits outweigh the costs, and there is an increasing availability of speech systems that not only transcribe text but also allow users to control computing applications and instruments using voice [3].

There are few speech-driven medical applications in production for recording speech and processing the resulting text into coded concepts. Two studies have evaluated the feasibility of this type of system. Lacson and colleagues [23] evaluated a system that records spoken dialogue between a hemodialysis patient and a nurse, automatically summarizing the conversation. Happe and colleagues [24] assessed the feasibility of mapping clinical concepts from speech-generated

dictations of discharge summaries to a standardized vocabulary. Additionally, the company M*Modal produces speech understanding applications and offers conversational speech and natural language understanding services to healthcare providers [25, 26]. Their speech understanding technology analyzes physicians' free-form dictation recordings and encodes clinical concepts and their modifiers and relationships into structured documents that can be imported into Electronic Health Record systems [25, 26]. Medical speech recognition technology has made considerable advancements in recent years, and some believe it will soon replace the need for human transcriptions in the field [5].

2.4 SPEECH RECOGNITION IN DENTISTRY

Speech applications in dentistry share some features with medical applications but also have distinctly different functionality [10]. As opposed to medical applications, dental speech applications typically implement command-and-control functionality as well as the transcription of free text. The command-and-control functionality supports two types of activities: (1) navigating within the application—for example, to select a specific patient—and (2) entering structured data in text fields, list boxes, radio buttons and checkboxes. Transcription is used primarily for dictating progress notes, surgical reports and similar textual information.

The speech recognition functionality of current dental software, when used with rigorous and meticulous attention to their implementation requirements, tend to work well because humans can easily adapt to the idiosyncrasies of the technology [27]. However, the ability to adapt obscures a crucial flaw in the design of most existing speech applications: dental speech applications do not communicate in the language of the user. As Yankelovich [28] and others

[29] point out, speech applications should not be built “on top of” graphical user interfaces because they force the peculiar language of programmers and software engineers on the user—for example, a user should not have to say:

> "Select Patient," "Sierra," "Mike," "India, "Tango," "Hotel," (for “Smith”) "Move Down One," (for moving to the second Smith in the search result) and "OK,"

when he would normally tell an assistant:

> "Please get me the chart for Mrs. Smith from Maple St."

(example produced using: Dentrix Chart Version 10.0.36.0; Dentrix Voice Version 4.0; Dentrix Dictation Version 2.0, all Henry Schein, Melville, NY).

To date, there is no comprehensive evaluation of the currently available speech recognition products in general dentistry. In this dissertation, we evaluated four leading dental practice management systems by having dental students chart patient exams via speech. We evaluated their accuracy, efficiency and the user satisfaction. Through this evaluation, we discovered the features, functionality and abilities of leading speech recognition systems in dentistry.

2.5 NATURAL LANGUAGE PROCESSING

The idea of automatically processing free-text and mapping information into a machine-understandable representation can lend itself to many useful applications. Natural language processing (NLP) involves processing, understanding, and analyzing information contained in free-text documents. NLP is used for classification, information extraction, summarization,

question answering and machine translation. In the following paragraphs, each of these NLP tasks is described.

Classification is simply a way of automatically categorizing information, and it can be done at the document, sentence, or word level. A classic example of classification is part-of-speech (POS) tagging. In POS tagging, words in documents are categorized according to their part of speech, such as noun, verb, adjective, or adverb. NLP systems can use parts of speech to further understand the information in the sentence—for example, interpreting the word "left" as an adjective or verb changes its meaning in the phrase "left upper lobe." Cohen et al. created a system that classifies discharge summaries according to psychiatric dangerousness which identifies patients that maybe a danger to themselves or others. [30].

Information extraction (IE) is the automatic extraction of specific information from free-text. There are many levels of IE determined by the goals of the final NLP application. For example, for the phrase "Ticlopidine 250 mg bid" the only item of interest may be what medicine was prescribed, but if dosage and frequency of dose are of interest, an IE application could also extract the dose of "250 mg" and the frequency of "bid." Many of the NLP applications applied to biomedical texts are IE applications that extract information such as diseases, patient information, and procedures from text [31-33]. One such system created by Xu et al. extracts patient demographic information from clinical trial abstracts to provide researchers with information about the trials. The application we developed and implemented in this dissertation is an IE application that extracts dental findings and restorations, along with their relevant modifiers.

Summarization systems interpret and summarize the main points of information found in free-text documents. This NLP technique can automatically produce a high-level view of a

document, saving the time needed to manually review a long document. For example, an NLP system that performs summarization may process a free-text clinical note and create sections that briefly summarize the patient history and current medications. Summarization involves not only extracting relevant information but also integrating the information and often generating natural sentences. Morioka and colleagues developed a system to structure free-text neuroradiology reports [34]. The summary report that is created from the free-text dictation presents data in the digital image and communication in medicine (DICOM) standard for structured reporting [34].

Question answering is a lofty goal of NLP that involves understanding questions, searching for answers, and generating text to address the question. An example of a question-answering application would be a system that answers clinician's questions about treatment of a patient. For example, a physician may ask a modified version of PubMed what the best antibiotic is for a patient with decreased renal function. Jacquemart and Zweigenbaum created a prototype designed to provide answers to questions posed by oral surgery students in which keywords were extracted from questions to identify relevant answers from websites [35].

Finally, machine translation is automatically translating from one language to another—e.g., from Spanish to English. Rosemblat and colleagues compared the effectiveness of two machine translation information retrieval methods that query information from the English-language website ClinicalTrials.gov and translate it into Spanish [36].

Because the work in this dissertation uses an information extraction application, the following discussion of NLP focuses on IE.

2.5.1 Linguistics

Many NLP techniques incorporate linguistic knowledge. Linguistics is the study of natural language and incorporates a number of sub-fields. These include the study of language structure with phonetics and phonology, morphology, and syntax, along with the study of meaning with semantics, pragmatics, and discourse. Linguistics in context of NLP is described in this section and was written using information from Jurafsky and Martin's book entitled "Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition." [15]

Phonetics is the study of language sounds and how they are physically formed while phonology is the study of sounds systems like syllables. Phonetics and phonology are used in NLP systems that convert speech-to-text or text-to-speech. Speech-to-text systems identify individual sounds and use dictionaries of word pronunciations and probabilities of combinations of sounds to translate sounds to words. For example, the sounds /th/ and /ə/ make up the word "the."

Morphology is the study of smaller units of meaning that make up words. For example, the word "disconnect" is made up of two smaller units, "dis" meaning *not* and "connect" meaning *to put together*. NLP can use morphological parsing to assign parts of speech (e.g., the suffix "-tion" can indicate a noun), to stem words to their root form (e.g., "walked," "walks," and "walking" all stem from "walk"), and sometimes to understand the meaning of words (e.g., in medicine the suffix "-ectomy" indicates a procedure in which something is removed from the body, such as a "splenectomy").

Syntax is the study of the structural relationship between words. Parsing is computing automatically the syntax in a sentence or phrase. Natural human sentences frequently contain a

multitude of ambiguity. Syntax can help humans understand the correct meaning of a sentence even if it is grammatically incorrect. For example, consider the grammatically incorrect sentence "Quickly runs Mary." In English-language syntax nouns precede verbs and adverbs modify verbs. As such, by knowing the parts of speech, we can apply syntax rules to rearrange the sentence correctly to "Mary runs quickly."

Semantics is the study of the meaning of words. NLP systems use semantics to map words to concepts and define how concepts relate to each other. Semantics can be used for word sense disambiguation and understanding meaning in a sentence. As seen in the example, "The stolen painting was found by the tree," the speaker's meaning of the word could support a tree having found a painting or the painting being located near the tree. Word sense disambiguation algorithms try to understand the meaning of words and select the most likely meaning given the context in which the word appears.

Pragmatics is the study of the speaker's meaning within the context of an utterance. Without knowing the context of the sentence "Tom saw the man with binoculars," it would be difficult to determine if Tom was using the binoculars to see the man or Tom observed a man holding binoculars. NLP uses pragmatics to discern meaning from the context of the free-text that it is processing.

The last linguistic component of NLP is discourse, which is the study of language units larger than a single utterance, such as related groups of sentences. One example of a task that requires knowledge of discourse is coreference resolution. Textual documents often contain expressions that refer to the same entity. For example, in the expressions "The patient fell down the stairs last night. She complains of a sharp pain in her elbow," "The patient" and "She" are referring to the same entity.

Many believe that for a computer to understand language the way a human does, linguistic knowledge is a necessary component of am NLP application [15]. The types of linguistic knowledge required depends largely on the task being performed. As described in Section 2.5.3, the NLP system we used in this dissertation uses both syntactic and semantic knowledge to interpret information from dental exams. Understanding both what the dentist is saying and the dentist's intents while dictating exams allows us to extract and chart the correct findings.

2.5.2 Typical NLP pipeline for information extraction

As an illustration of the way NLP systems work, we describe a typical IE pipeline. This description is modeled after a few of the first publicly available NLP applications for clinical texts: the HITEx (Health Information Text Extraction) system [37] and cTAKES [38]. Both of these systems use a pipeline of modules for information extraction, which can be a useful way to describe the elements involved in building an IE system.

One of the first tasks for an IE system is sentence segmentation. Text has to be broken down into understandable chunks to be processed, and in natural language sentences are natural constituents. The next task is word tokenization—splitting sentences into tokens or words. Tokenization is sometimes very challenging, because some words contain punctuation that can be confused with sentence delimiters, as in "b.i.d."

Once words are defined, they can be tagged with the appropriate part of speech. Part of speech (POS) taggers automatically assign part-of-speech tags to each word in a sentence. HITEx uses a rule-based POS tagger [37]. The next stage of the NLP pipeline is a syntax parse. Systems can perform deep or shallow parses. Shallow parses only assign phrasal categories to

simple phrases, such as noun phrases and verb phrases, whereas deep parses attach phrases to each other to build more complex—or deep—phrasal structures. Syntactic parsers require two elements to assign phrasal categories to sentences: a lexicon defining which parts of speech each word can be assigned and probabilistic or rule-based grammars that indicate which combinations of words and phrases are allowed in the composition of sentences and phrases.

These modules in the pipeline are typically used to assist an IE system in assigning meaning to words, phrases, and sentences. Relevant information from the text can be semantically classified or mapped to a standardized vocabulary. More detailed IE systems can map the information from sentences to templates that indicate semantic classes and their modifiers. Many IE systems in biomedicine map words and phrases to UMLS (Unified Medical Language System) concepts—for example, "pain in the chest" can be mapped to the UMLS concept "Chest Pain" (UMLS C0008031).

Once concepts are identified and mapped to a standardized vocabulary, a typical IE application takes into account contextual information like negation, uncertainty, and temporal information. An important aspect of understanding the meaning of a clinical report is knowing that in the sentence "denies chest pain," the concept "chest pain" is negated. There have been several algorithms developed for identifying negation in biomedical texts [39, 40].

Once concepts are identified, an IE system may identify relationships between concepts. For example, in the sentence "The gold crown on tooth 2 is fractured," the dental condition concept "fracture" is actually at the location of the dental restoration concept "crown." Identifying relationships among concepts is critical to understanding the text in a way that is useful for applications of NLP.

Finally, an IE pipeline may use discourse processing to help with tasks like topic segmentation and coreference resolution. For example, a blatant discourse marker like, "that concludes the physical exam" can indicate the end of a section in a report.

NLP pipeline systems use various modules for information extraction, and the system developers apply a broad range of methods to accomplish each of the modules' tasks, described in the next section.

2.5.3 NLP Methods

NLP systems can employ either symbolic or statistical methods, or sometimes both. Systems that use symbolic approaches focus on linguistic knowledge as described in Sections 2.5.1 and 2.5.2. Symbolic NLP systems use methods like regular expressions for tokenization, rules for POS tagging and grammar rules and lexicons for syntactic parses.

Statistical NLP systems complete the same tasks as symbolic NLP systems but use statistical methods. An example is a statistical bag-of-words model which can be used for document classification. In statistical bag-of-words models, the text is parsed and a collection of tokens or n-grams is created disregarding grammar and word order. The collection of tokens is statistically relevant to the text used to create it. As such, the tokens found in a bag-of-words from a set of radiology reports could allow a system to classify a report as a radiology report when compared to a pathology report. Probabilistic models are another statistical technique that can assist with the identification of concepts and relationships. A probabilistic model could be trained to learn that "caries" and "cavity" both refer to the concept "caries." The model will make decisions on the words "caries" and "cavity" when they occur in new text documents based on the probabilities of occurrence found in the training data.

Finally, there are hybrid systems that use both symbolic and statistical methods for information extraction. ONYX, the NLP system used in this research is a hybrid system [41]. ONYX semantically interprets each relevant phrase before a syntactic parse is completed. It uses training data probabilities to score each phrase's semantic interpretation and the score guides a semantically guided best-first search for filling in a semantic template with concepts and relationships [41]. This hybrid method allows ONYX to completely parse relevant phrases as the system finds them.

Using NLP techniques to successfully process a dictated dental exam and match words from the dentists with their underlying concepts is the first step to developing a natural language system for dental charting.

2.6 NLP IN MEDICINE

Over the last few decades the medical informatics community has applied NLP techniques [42, 43] to a variety of biomedical domains, including radiology [44-52], emergency medicine [53], pathology [54, 55], public health [56-61] and oral surgery [35].

NLP techniques have been used in a wide range of medical informatics applications including quality assessment in radiology [62, 63], identification of structures in radiology images [32, 64], facilitation of structured reporting [34, 65] and order entry [66, 67], encoding findings required by automated decision support systems such as guidelines [68] and antibiotic therapy alarms [31], identifying patient cohorts from physician notes [69], classifying discharge summaries according to psychiatric dangerousness [30], processing spoken dialogue from home hemodialysis phone calls [23], identifying research subjects from the electronic medical record

[70, 71], constructing computable eligibility rules for clinical trials [72], assigning ICD-9 codes to radiology reports [73], identifying smoking status from medical discharge summaries [74], diagnostic support [75, 76] and improving public access to medical knowledge [33].

Most research on medical language processing applications applied to clinical text has focused on identifying instances of targeted conditions at the sentence or phrase level. Integration of the individual instances has not been well addressed as often in medical informatics research. As an exception, the system MedSyndikate [77] incorporates discourse processing and coreference resolution to extract information from pathology reports. MedSyndikate uses a list of discourse markers to establish and keep track of reference relations between words [78]. Also, Rindflesch and Fiszman described a methodology which focused on identifying semantic relations among individual concepts in text [79]. They use information from the Unified Medical Language System to help identify related concepts called hypernymic propositions that appear frequently in scientific text [79]. Identifying the concepts and their relationships helped them distinguish between new and old information in the text [79].

NLP techniques have performed quite well within limited domains. One study showed that an NLP system called MedLEE could identify radiological findings, such as atelectasis, pulmonary masses, and infiltrates with an average sensitivity of 81 percent and specificity of 98 percent [49]. Another study showed an NLP system called SymText could identify several findings consistent with acute bacterial pneumonia with sensitivities ranging from 84 to 95 percent and specificities around 98 percent [31]. In both of these studies, the NLP system's performance was not significantly different from physicians' performance.

2.7 DEVELOPING & EVAULATING SEMANTIC REPRESENTATIONS

Section 2.7 was accepted for publication in AMIA 2009[80] and is reproduced with permission from the original publisher, in its entirety, with minor revisions. The following is a list of author contributions: Irwin: *wrote manuscript, model developer, intellectual contributions, annotator; Harkema*: *wrote manuscript, model developer, intellectual contributions, annotator; Christensen*: *wrote manuscript, model developer, programmed NLP system, intellectual contributions, annotator; Schleyer*: *intellectual contributions, domain expert; Haug*: *intellectual contributions; Chapman*: *wrote manuscript, model developer, intellectual contributions.*

NLP applications that extract information from text rely on semantic representations, like semantic networks [81], to guide the information extraction (IE) process and provide a structure for representing the extracted information. Semantic representations model the concepts and relationships that are important for the target domain and that appear in the relevant document collections. The structure of semantic representations must support further processing of the extracted text required by the final NLP application and are thus constrained by the capabilities of the NLP engine driving the application.

Since the content of a semantic representation depends largely on a document set and an application, it is usually not possible to "plug in" a previously developed semantic model. Also, existing domain ontologies are less useful as a model for structuring the information found in actual text because they tend to focus on abstract descriptions of knowledge organization. Therefore, it is often necessary to build a new semantic representation as part of an IE project. Although there is some documentation about the evaluation of semantic networks [82], there is no widespread literature concerning the detailed process of constructing semantic representations for NLP applications. In the context of an IE project, we devised a four-step methodology for

developing and evaluating semantic representations. The methodology integrates principles and techniques in semantic modeling, annotation schema development, and human inter-annotator evaluation.

2.7.1 Methods

We created a four-step methodology for developing and evaluating a semantic representation that integrates the following: principles for the creation of semantic representations, methods for the development of annotation guidelines and schema, and methods for evaluating semantic representations base on inter-annotator agreement. The four steps include: (1) develop an initial representation from a set of training texts, (2) iteratively evaluate and evolve the representation while developing annotation guidelines, (3) evaluate the ability of domain experts to use the representation for structuring the content of new texts according to the guidelines, (4) evaluate the expressiveness of the representation for information needed by the final application.

In creating and evaluating our representation, we wanted to address five standard requirements for a semantic representation [15]: (1) verifiability: the ability to validate statements from the represented knowledge; (2) unambiguous representations: a representation with only one valid interpretation, but is able to support a level of vagueness; (3) canonical form: inputs that have the same meanings should have the same representation; (4) inference: the ability to infer information not explicitly modeled; and (5) expressiveness: the ability to model unseen but relevant information. In this section, we describe the first two steps of the methodology, using a case study from our experience of modeling chartable information from a dictated dental exam.

STEP 1: DEVELOP AN INITIAL SEMANTIC REPRESENTATION. The first step in developing an initial semantic representation is to determine which concepts to extract and model. This decision is largely driven by the final application and the feasibility of automated extraction. For our study, we identified the 13 most frequently occurring dental conditions: filling, crown, caries, missing tooth, abrasion, sealant, bridge, denture, root canal, fracture, veneer, inlay, and attrition.

We created our semantic representation using a bottom-up, data-driven approach. In this approach, one uses the textual source of information—in our case dictated dental exams—to design a representation for the mappings from words to concepts, as well as the relationships among the concepts.

To create our representation, we read a single transcribed dental exam—containing 551 words—and identified the information in the text related to the 13 target conditions. To represent the information in the exam, we created two types of semantic representations: a semantic network and concept models (CM).

For each statement in the exam, we identified any concepts related to one of the 13 dental conditions. For example, in the sentence "There is a cavity on tooth 2," we identified two concepts: a dental condition of caries and an anatomic location of tooth 2. We developed a CM with non-terminal nodes for the concepts and terminal nodes for the words from the text that indicated the concepts, as shown in Figure 2. We then labeled relationships among the nodes.

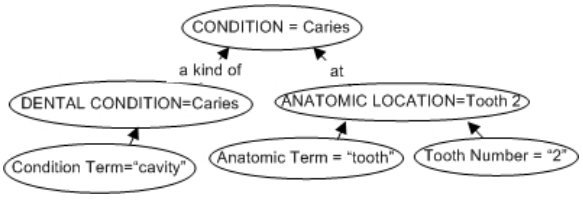

Figure 2. Initial network from training sentence "There is a cavity on tooth 2."

It became clear that we did not only need a CM for the way words are used to describe concepts, but we also needed a mechanism for relating concepts to each other. For example, the statement "crack on the crown of tooth 2" describes three concepts: a DENTAL CONDITION called fracture, a RESTORATIVE CONDITION called crown, and an ANATOMICAL LOCATION called tooth 2. Understanding the relationship between the crack and the crown is critical to our ability to chart the information. Therefore, we developed a semantic network encoding general domain knowledge to represent allowable relationships among dental concepts (Figure 3). Terminal (white) nodes in the semantic network represent the root of individual CMs. Nonterminal (gray) nodes represent abstract types with no associated CMs that are useful for indirect relations and discourse processing. The semantic network allows different types of relationships between concepts—for example, the network expresses the relations *at*(CONDITION, ANATOMIC LOCATION) and *has*(ANATOMIC LOCATION SURFACE). The semantic network also represents taxonomic relationships, via the *a kind of* label. A type may have multiple parent types—for example, RESTORATIVE CONDITION is a subtype of both CONDITION and LOCATION.

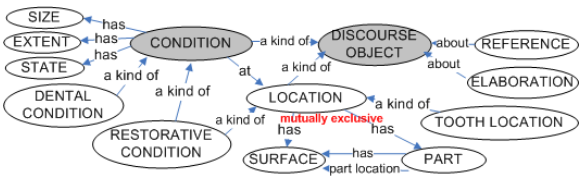

Figure 3. Semantic network for our domain. White nodes represent the top node in an independent concept model. Arrows represent relationships among the nodes.

Figure 4 shows how we use both the semantic network and CMs to interpret the sentence "Fifteen has one occlusal amalgam." We infer concepts from values in the leaf nodes of the CMs and then use the semantic network to model the relationships among the inferred concepts.

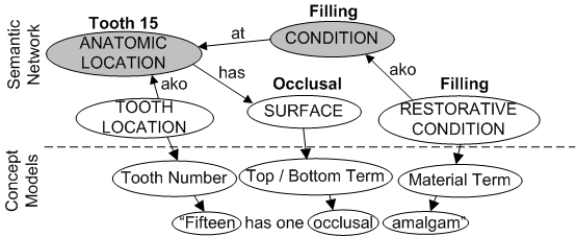

Figure 4. Example of the ideal interpretation of the sentence "Fifteen has one occlusal amalgam." Words above nodes are the inferred concepts.

STEP 2: EVALUATE AND EVOLVE THE REPRESENTATION AND DEVELOP ANNOTATION GUIDELINES. Step 2 is an iterative process involving structuring information from new documents to evaluate the coverage of the current representation, to evolve the representation based on new data, and to develop or enrich guidelines to ensure consistency among annotators.

We selected 12 exams of new patients: one exam from our original dentist, six from a new dentist and five from a hygienist. We developed a training tool for assisting human annotators in structuring information from a report into the concept networks. Three annotators—an informaticist, a linguist, and a developer—with input from dental experts, independently reviewed two exams identifying any instances of the 13 target conditions and related concepts found in the exams. The annotators entered the terms from the exam into the terminal nodes of the CMs—for example, for the sentence in Figure 2, the word "cavity" was slotted in the condition term node, the word "tooth" in the anatomic location node, and the word "2" in the tooth number node. The annotators created values for the non-terminal nodes (i.e., implied concepts) —for example, in Figure 2, the dental condition node received the value Caries, and the anatomic location node Tooth Two. According to the semantic network, the training tool generated all allowable relationships between instantiated CMs for that sentence, and each annotator selected the semantic relationships for each related pair of CMs. The sentence

in Figure 2 has two relevant relations: *at*(CONDITION, ANATOMIC LOCATION) and *akindof*(DENTAL CONDITION, CONDITION).

After structuring the information from two exams, the three annotators met to discuss disagreements, to come to consensus on the best instantiations, to change the CMs or semantic network—in order to successfully model the information in the two exams—and to clarify the guidelines. The annotators iterated through the set of 12 reports in six cycles, annotating two reports independently before each meeting.

After each iteration, we measured agreement between pairs of annotators. Because it is not possible to quantify the number of true negatives in text annotation, we could not use Kappa. Therefore, we calculated agreement via inter-annotator agreement (IAA) [83]. IAA= matches / matches + non-matches, where matches = 2 x correct, and non-matches = spurious + missing. We calculated IAA separately for words, concepts, and relationships. Step 2 can be repeated until agreement reaches a threshold level or plateaus and the models appear stable and complete.

2.7.2 Results

We developed initial models using a single report of 551 words and evolved the models through iterative cycles of independent annotation and consensus meetings. Our final model resulted from annotations of 289 sentences in 13 reports.

DEVELOPMENT OF INITIAL MODELS. We identified 33 sentences containing relevant conditions—hereafter called cases—in the training exam. From those 33 cases we instantiated 125 words (73 unique) and 160 concepts (74 unique) into the CMs. Our initial semantic network had 11 nodes, eight of which represented individual concept models. After annotating the 12 exams in six iterations, changing the semantic model and concept models to

accommodate all relevant information in the exams, the semantic model contained 13 nodes, 11 of which were concept models and 15 relationships. (see Figure 3).

Because we used a data-driven approach to design the initial models, we revised them several times to account for new concepts described in unseen exams. One type of change was modularizing the CMs. Having a semantic network removed the need to link related concepts within large CMs, so we, for example, split the ANATOMIC LOCATION and DENTAL CONDITIONS networks shown in Figure 2.

We added nodes to CMs and the semantic network and added new CMs. For example, although initially we attempted to use the same CM for dental conditions (caries and fractures) and restorative conditions (crowns and fillings), we ultimately created separate DENTAL CONDITIONS and RESTORATIVE CONDITIONS networks, because we found these conditions have different properties.

We also added new relationships to the semantic network to capture the different roles the same concept can assume in different contexts—for example, the word "crown" can indicate a restorative condition (“crown on 16”) or the location of a dental condition (“fracture on the crown.”)

EVALUATING AND EVOLVING THE MODEL. Generally, as annotators instantiated cases, they found that a case consisted of a dental or restorative condition at an anatomic location. In the 12 exams two or more annotators identified a total of 256 cases for an average of 21 cases per exam. Further, for the 256 cases, each annotator slotted an average of 783 words and 1018 concepts and defined an average of 394 relationships.

The average agreement for the three annotators for all iterations was 88 percent: 88% for words, 90% for concepts, and 86% for relationships—figure 5 shows the average IAA for each

iteration. All changes to the CMs and semantic network occurred after iterations one through four, but we made no changes after iterations five or six.

Disagreements among annotators can reveal lack of expressiveness and ambiguity in the semantic representations. For example, annotators slotted "some" in "22 has some incisal wear" in the severity term node, which is a modifier in the CONDITION CM. However, annotators disagreed on where to slot the similar word "small." In the end, we created a new CM for size.

Disagreements can also indicate inadequate annotation guidelines. After each iteration, we changed the annotation guidelines based on our discussions of how to best model the concepts in the text. IAA dropped in the second iteration due to multiple cases in which the annotators disagreed on how to slot the words "not missing" and "not present" —as seen in the sentence "tooth number one is not present." We made almost half (8/20) of the changes to the guidelines during the discussion after iteration 2.

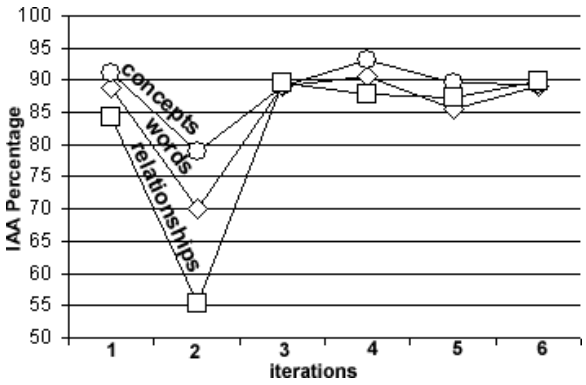

Figure 5. Graph of average IAAs for each iteration.

A key benefit of the iterative annotation phase is to enrich the guidelines while developers perform annotations so that the guidelines presented to experts in Step 3 will be as clear and useful as possible.

2.7.3 Discussion

As we began developing a semantic representation for our NLP system, we searched the literature for advice on how to best create a semantic model and on how to determine its quality. Although we could not find articles directly addressing development and evaluation of semantic models, we found relevant techniques in related areas, which we integrated in a four-step methodology we have begun to implement.

The methodology addresses principles for the creation of semantic representations [15], including a model's expressivity, its ability to represent information unambiguously, and the ability to map information to canonical form. The methodology incorporates techniques used in training annotators to develop training and testing sets for assessing output of an NLP system. Our method is similar to Roberts and colleagues [83] who compiled an annotated corpus of clinical reports, trained annotators on a semantic network they developed and iteratively evaluated agreement.

The first step of the methodology—creating the representation from example documents—allows developers to design models that relate the words in the text to the meaning conveyed by the words. To our surprise, creating our initial representations from a single document took several months as our models changed multiple times in an attempt to model what the dentist said in the exam.

The second step—iteratively evaluating the representation by annotating new documents—is a critical step for ensuring generalizability of the models and for writing annotation guidelines to help non-developer annotators. This step is a quantitative step that allows developers to measure agreement and reveals deficiencies in the existing models. While

slotting cases in Step 2, annotators test the representation's expressiveness and ability to support unambiguous representations while assigning words to canonical form.

The third step—evaluating agreement among expert annotators who follow the guidelines—is a familiar step in assessing the quality of training and test set annotations that serves a second purpose: to determine how usable the models are by non-developers. Our representation is quite complex, and we look forward to measuring its usability by dentists.

The fourth step—evaluating the expressiveness of the representation for information needed by the final application—is important for determining whether the models really convey the same information conveyed by the text. We did not complete this step as a part of this study and it is outside the scope of this dissertation. However, when we do complete this step, we plan to use the methodology described by Rocha et al. [82]. For this step, we will present domain experts—dentists, in our case—with two types of exams: transcriptions of dental exams for one set of patients and semantic models with manually instantiated information from the exams for another set of patients. We will test the ability of the domain experts to answer questions based on the two exam formats (in our case, the experts will graphically chart the exam). If the semantic representation successfully conveys relevant information from the text, the experts should answer questions from the semantic representation as well as from the text itself.

Our approach is a largely bottom-up approach, which can be an effective method for designing models for representation of ideas expressed in text. Disadvantages of a bottom-up approach include not leveraging expert knowledge contained in existing models and the possibility of designing a model that can only be used for a specific task. When we began development, we explored the UMLS and the Foundational Model of Anatomy as potential models; however the UMLS dental entries were limited, and existing dental concepts did not

map well to what we saw in dental exams. The Foundational Model of Anatomy described relationships between dental anatomical structures; however it did not contain information pertaining to dental conditions or restorations. In spite of using the text to drive our model development, we frequently consulted with dentists to ensure our models were consistent with domain expertise.

In this section, we described a process for developing and evaluating a semantic representation for an NLP application and illustrated the process in the domain of spoken dental exams. The methodology we describe explicitly addresses general requirements for semantic representations using a data-driven and iterative approach that can be replicated by others. As described, we carried out the first two steps of the methodology, illustrating the types of changes we made to our models through our approach. Although we applied the methodology to a single domain, the methodology is based on standard principles and approaches that are not dependent on any particular domain or type of semantic representation.

2.8 ONYX

Excerpts from Section 2.8 were published in [41] and are reproduced with both the author's and the original publisher's permission.

We developed the semantic model for an NLP system called ONYX [41]. ONYX is a sentence-level medical language analyzer that processes a transcribed dental exam and maps its content onto the semantic models described in Section 2.7. ONYX uses two types of semantic models: a semantic network that models the concepts we want to extract and their relationships to each other; and concept models to slot the words extracted from the exam [41]. For example,

for the sentence “there is a mesial cavity on tooth 2”, ONYX slots the values “mesial”, “cavity” and “tooth 2” in the leaf nodes of concept models and then uses the semantic network to model the relationships among the inferred concepts [41]. Each concept is represented by a template as shown in Figure 6 [41].

Dental Condition	
Condition	*caries
Condition term	“cavity”
ToothLocation	
Tooth Location	*numberTwo
Tooth Number	“2”
Surface	
Surface	*mesial
Front/Back Term	“mesial”

Figure 6. ONYX templates for “There is a mesial cavity on tooth 2.” From [41], used with permission.

A template consists of a set of slots filled with either a phrase taken directly from a sentence in the text or a term that is inferred from phrases or terms elsewhere in the template [41]. Allowing inferences within a template produces a consistent representation of the meaning of a sentence by abstracting away from different ways concepts and relationships can be expressed in text [41]. Assignment of phrases and terms to slots in a template is based on probabilistic models similar to the Bayesian Networks used in MPLUS [84] (a probabilistic medical language understanding system) and SYMTEXT [85] (a natural language understanding system for encoding free text medical data). However, the templates in ONYX employ a more efficient model of computation [41]. The probabilistic models are derived from a training set of annotated documents [41]. Currently, ONYX has been trained on 13 exams from two dentists and one hygienist. ONYX’s processing is strongly guided by both syntactic and semantic constraints as described in more detail in [41].

ONYX's output for a given sentence is a conjunction of binary predicates [41]. The predicates correspond to the relationships specified in the semantic network and the predicates' arguments correspond to inferred terms in the concept networks [41]. Asterisks in the predicate indicate that the arguments are inferred terms—for example, the ONYX interpretation of the sentence "There is a cavity on tooth 2" is [41]:

CONDITIONAT(*CARIES, *NUMBERTWO) & LOCATIONHASSURFACE(*NUMBERTWO, *MESIAL).

ONYX has only been trained on hard tissue examinations. The version of ONYX used in this dissertation utilizes a semantic model that was designed based on 13 of the most common hard tissue findings: filling, crown, caries, missing tooth, abrasion, sealant, bridge, denture, root canal, fracture, veneer, inlay, and attrition. Therefore, the charting abilities of our speech-to-chart prototype will be limited to these findings.

3.0 RESEARCH OBJECTIVES

The objectives of this dissertation were to: (1) evaluate the efficiency, effectiveness, and user satisfaction of the speech interfaces of four existing dental practice management systems; and (2) to develop and evaluate a speech-to-chart prototype for charting naturally spoken dental exams.

3.1 RESEARCH OBJECTIVE 1: EVALUATE THE EFFICIENCY, EFFECTIVENESS, AND USER SATISFACTION OF THE SPEECH INTERFACES OF FOUR EXISTING DENTAL PRACTICE MANAGEMENT SYSTEMS.

3.1.1 Motivation

Four of the leading dental software packages—which encompass over 80 percent of the market—provide a speech interface for data entry. Our preliminary data have shown that these speech interfaces are cumbersome to use and poorly designed. However, to date, there is no comprehensive evaluation of the currently available speech recognition products in general dentistry. With an in-depth evaluation of the leading systems, we can discover the features, functionality and abilities of leading speech recognition systems in dentistry.

3.1.2 Research Question

Is the current state of speech recognition for charting in dental software systems insufficient for use during initial dental exams?

3.2 RESEARCH OBJECTIVE 2: DEVELOP AND EVALUATE A SPEECH-TO-CHART PROTOTYPE FOR CHARTING NATURALLY-SPOKEN DENTAL EXAMS.

3.2.1 Motivation

Current speech interfaces of dental practice management systems are poorly designed and cumbersome to use. The absence of a flexible, robust, and accurate natural language interface is a significant barrier to the direct use of computer-based patient records by dental clinicians. To enhance clinical care we developed and evaluated a speech-to-chart prototype to support the flexible and familiar communication style inherent in the natural language dictation of hard tissue dental examinations.

3.2.2 Research Question

Can speech recognition and natural language processing be used to create a prototype digital charting system that performs with accuracy similar to that of existing dental practice management systems?

4.0 OBJECTIVE 1: EVALUATE SPEECH FUNCTIONALITY IN DENTAL PRACTICE MANAGEMENT SYSTEMS

To date, there is no comprehensive overview or evaluation of the currently available speech recognition products in general dentistry. To address objective 1, we compared: (1) the speech-interface features and functions of existing practice management systems, and (2) the efficiency, accuracy and user satisfaction of the speech interfaces of each system.

4.1 FEATURE AND FUNCTION COMPARISON

Excerpts from Section 4.1 were published in [8] and are reproduced with the permission of the original publisher.

To summarize the existing availability of speech interfaces for dental data entry, we compared the speech functionality of four dental practice management systems (PMS): (1) Dentrix v.10 (Henry Schein, Melville, NY), (2) EagleSoft v.12 (Patterson Dental, Effingham, IL), (3) PracticeWorks v.6.0.5 (Kodak, Atlanta, GA) and (4) SoftDent v.11.04 (Kodak, Atlanta, GA) [8]. These four systems make up approximately 80 percent of the current practice management market in the United States [8]. For this comparison, we explored each system and created a checklist to highlight and compare the speech interface features that are present in each system [8].

4.1.1 Methods

First, we acquired full working versions of the four PMSs and installed each program in its default configuration [8]. We reviewed each program's user manual to learn about the system's speech features and functions [8]. We then manually explored and used each system's speech interface features focusing on the clinical components [8]. Last, we contacted the software vendors to answer any specific questions regarding the system's speech functionality and features [8]. As we explored each system, we created a comparison checklist to highlight each program's speech interface capabilities including which features were present or absent [8].

4.1.2 Results

Table 1 presents our comparison checklist of the speech interface features of each system [8]. Salient findings include that Dentrix and EagleSoft used the Microsoft Speech Recognition Engine (Microsoft, Redmond, Wash), whereas PracticeWorks and SoftDent used the default speech engine installed on the computer as long as it had SAPI (Speech Application Programming Interface) version 4.0 or 5.0 program files [8]. Dentrix was the only program that allowed free text dictation into a "clinical notes" area [8]. However this feature required the purchase and installation of Dragon NaturallySpeaking (Nuance Communications, Burlington, MA) [8]. The speech training sessions for all of the systems took approximately five to ten minutes to complete [8]. To use the free text dictation in Dentrix, an extra 30 minutes of training was necessary [8]. All four of the systems allowed a user to complete extra training if necessary, and EagleSoft, PracticeWorks and SoftDent allowed the user to train with specific dental terms [8]. Next, none of the programs allowed hard tissue and periodontal charting with naturally

spoken dictations—that is, all required specific speech commands [8]. The number of specific speech commands available to interface with the software varied across the systems: Dentrix had approximately 573, EagleSoft had 140, SoftDent had 53 and PracticeWorks had 41 [8]. Only Dentrix allowed the use of the international communications alphabet (e.g., alpha, bravo) to assist with interactions [8]. All four systems had the ability to provide audio confirmation (i.e., feedback) of a given command, but only EagleSoft and PracticeWorks gave complete visual confirmation of commands [8]. For example, Dentrix repeated a command after a user spoke it while EagleSoft displayed the transcribed command on the screen. Dentrix and SoftDent did not provide visual conformation for some actions in their default installation [8].

Table 1. Functions that can be completed via speech. Adapted from [8] with permission.

	Systems			
	Dentrix	**EagleSoft**	**PracticeWorks**	**SoftDent**
hard tissue charting	Yes	Some	No	No
periodontal charting	Yes	Yes	Yes	Some
dictate raw clinical notes	Yes	No	No	No
chart existing and proposed findings	Yes	Yes	No	Some
select tooth surface	Yes	Some	Yes	Some
select patient	Yes	Some	No	No
open chart	Yes	Yes	Some	No
select items from list via name shown	Yes	Some	No	No
navigate through chart (next, move down two, etc.)	Yes	Yes	Some	Some
use all displayed options/buttons	Yes	Some	Some	Some
access menus, buttons, pop-ups, and checkboxes	Yes	Yes	Some	Some
undo last command	Yes	Yes	Yes	Some
clear/delete entries	Yes	Yes	Some	Some
start and stop listening via speech	Yes	Yes	Yes	Yes

4.1.3 Discussion

This comparison demonstrates that dental PMSs are attempting to accommodate clinical data entry via speech [8]. However, the existing systems' speech interfaces have many limitations that may hinder their use [8]. As shown in Table 1, speech functionality varies across all systems with PracticeWorks and SoftDent not having the ability to complete hard tissue charting via speech [8]. The fact that all systems required interaction in the form of specific speech commands demonstrated that these systems are not designed to be used without prior understanding of the software and the memorization of, or easy access to an enormous amount of specific terminology [8]. Further, the command-and-control functionality of the speech interfaces requires the clinician to consistently look at the screen not only to verify the correct items are being selected, but to select the next item—for example, in the case of a list where they have to say "move down four" to select the item of interest [8]. If speech commands were less command-and-control and supported a natural dictation flow and vocabulary, ease of use could be significantly improved [8].

Overall, speech interfaces of dental PMSs are somewhat cumbersome to use and poorly designed [8]. Limited speech functionality for clinical data entry has the potential to reduce the ability of clinicians to interact directly with the computer during clinical care. These issues could explain the limited use of speech interfaces in dentistry and the desire for improvements in this area [7]. In the future, dentistry will see the influx and be able to reap the benefits of decision support tools and shared electronic medical records [86]. The limitations of these speech interfaces can hinder direct computer use during clinical care which can in turn impede the benefits and effectiveness of any electronic patient records and clinical decision support systems.

4.2 PERFORMANCE EVLAUATIONS

Our feature comparison (Section 4.1) highlighted available speech interface functionality of existing practice management systems, but it did not evaluate the performance of these features. The objectives of the performance evaluations were to evaluate the efficiency, effectiveness, and user satisfaction of the speech interface functionality of the four previously reviewed dental practice management systems (PMS). To evaluate the systems, participants completed the same charting tasks in each of the four PMSs. We then evaluated the efficiency, accuracy, and user satisfaction of each system. This evaluation allows us to answer the questions: How long does it take to chart findings via speech in existing PMSs? How many and what types of errors do PMSs speech interfaces make? And how do users feel about these systems?

4.2.1 Methods

PARTICIPANTS. A convenience sample of 20 dental students from the University of Pittsburgh School of Dental medicine participated in this study. The sample of dental students was defined as convenient, because the students were not randomly selected to participate—they simply responded to recruitment emails and flyers. Participants with previous experience using clinical charting speech features in each PMS were excluded from the study. The Institutional Review Board at the University of Pittsburgh reviewed and approved this research protocol (IRB# 0610017).

SPEECH CHARTING TASK. For this study, we asked participants to verbally chart a simulated patient in each of the four Practice Management Systems (PMS). First, we created a simulated intraoral patient record that contained 18 different hard and soft tissue findings of the

maxilla. Second, we developed the verbal command scripts necessary to chart the simulated patient in each PMS. Because training the students to learn each complex and esoteric command was outside the scope of this project, we created a step-by-step script for each system made up of the commands necessary to chart each finding—table 2 shows excerpts from two of the scripts. The scripts demonstrate that in all of the systems very specific commands are necessary for charting. The speech commands are directly comparable to using a mouse. For example, to chart a mesial caries on tooth 3 a dentist must say: "conditions," "move down 9," "move down 8," (for moving to the caries item on a list of conditions) "OK," "mesial," "OK," "existing." (example produced using: Dentrix Chart Version 10.0.36.0; Dentrix Voice Version 4.0; Dentrix Dictation Version 2.0, all Henry Schein, Melville, NY). We sent each script we developed to its corresponding manufacturer to be evaluated for correctness and efficiency and incorporated all manufacturer edits into our final scripts.

Table 2 Excerpt from two scripts to recommend a B composite veneer on tooth 8.

Dentrix	**EagleSoft**
"select eight"	"tooth eight"
"procedures"	"quick pick menu eleven"
"move down four"	"menu item four"
"next field"	"proposed"

In the completed scripts, PracticeWorks and SoftDent were only able to chart periodontal findings via speech. Therefore, participants were only able to chart nine of the 18 findings in those systems—table 3 compares the scripts across systems. In three of the scripts it was necessary for the user to utilize the mouse or keyboard at some point during the task completion—that is, the charting could not be completed via speech alone.

Table 3. Comparison of commands necessary to complete the charting tasks as documented in the scripts. (H) – Hard tissue charting, (P) – Periodontal charting. PracticeWorks and SoftDent cannot chart hard tissue findings. Adapted from [8] with permission.

	Systems			
	Dentrix	**EagleSoft**	**PracticeWorks**	**SoftDent**
total number of commands in script	92	75	24	33
total number of speech commands in script	69 (H) 23 (P)	41 (H) 30 (P)	2 (H) 19 (P)	0 (H) 24 (P)
total number of mouse/keyboard commands in script	0 (H) 0 (P)	4 (H) 0 (P)	3 (H) 0 (P)	3 (H) 6 (P)
percent completed with speech alone	100	95	88	73

PERFORMANCE TESTING. To evaluate the efficiency and accuracy of the charts generated with PMSs, participants performed a charting task on each system. Each student scheduled four sessions—one for each PMS. When they arrived for their first session, participants were randomly assigned to a PMS. Once five students were assigned to the same system for their first session, we removed that system from the pool of possible systems to be tested during the first sessions. On the second visit, the student was randomly assigned to one of the remaining three systems and again once five students were assigned to a system it was removed from the pool of possible systems to be tested in the second visit. We repeated this procedure for each round of sessions (1-4). In this way, we attempted to control ordering affects—that is, each group of five students tested the systems in a random order. We required at least a two-week waiting period between sessions to reduce any memory or learning effects that may stem from using the same patient chart and similar speech commands in some systems.

To start the session, we administered a background questionnaire to each participant. The questionnaire was a modified version of a validated tool that measures dental students' use of, knowledge about, and attitudes towards computers [87]. We used the questionnaire to acquire a descriptive analysis of the participants' ages, sex, native language (English or non-English), prior computer experience, and affinity towards computers.

To complete the charting task via speech, all PMSs required a brief training session. This training was a component of the speech module for each system and was used by the system to learn how the participant pronounces words and other features of the participant's speech. This was not a comprehensive training of the charting software. We supervised participants during their training session to assist with problems and questions, to ensure that the headset microphone was adjusted properly, and ensure that the user was speaking optimally for the task. Successful completion of the training session was required to continue with the task evaluation.

After training, we asked the student to read verbatim the charting script corresponding to the system being tested. During the charting task, we assisted with problems that arose. We observed three common problems that required our attention or advice during the charting task. First, the system's response to a spoken command resulted in the student being off script—e.g., the chart would close/exit. In this case, we interrupted the participant, corrected the problem, and had the participant begin again either where they left off (if possible without redoing steps) or on the next finding. Second, the system did not respond at all to a spoken command. In this case, we asked the participant to repeat the command two more times (a total of three times) and then to either move on to the next command or—if that was not possible—move on to the next finding. Third, the system charted a finding incorrectly, but the participant was able to continue with the script—e.g., selecting the wrong tooth. In this case, we asked the participant to ignore the error and continue. The author took hand-written notes during each session. Also, we video recorded all sessions capturing the screen, including mouse clicks, and audio. The video was used as a reference during data analysis if the hand-written notes did not provide enough information.

At the conclusion of each session we asked participants to complete a user satisfaction questionnaire. The questionnaire—Figure 7—contained six open-ended questions designed to

gauge the participant's satisfaction of the system. Originally, the questionnaire contained 27 items based on the Subjective Assessment of Speech System Interfaces (SASSI) project [Hone, 2000 #169]. However, during our feasibility studies [8], many participants had difficulty answering the questions due to their limited script-based interaction with the system. Therefore, we modified the SASSI questionnaire to contain the set of six questions shown in Figure 7.

1. What, in your mind, were good things about the system you just used?
2. What, in your mind, were bad things about the system you just used?
3. Did this system perform according to your expectations about voice-activated systems in dentistry? If yes, how so? If not, why not?
4. What would be an acceptable repetition frequency for you if you would use such a system in your routine clinical work (one word per sentence, one word per patient exam, one word per day, ...)?
5. Would you be willing to enunciate your speech more clearly in order to use such a system in your routine clinical work?
6. Do you think you could enter clinical data faster with a system like this?

Figure 7. Participant satisfaction questionnaire.

DATA ANALYSIS. We evaluated efficiency, accuracy, and user satisfaction of the four PMSs. To measure efficiency, we analyzed the amount of time required to complete training and the amount of time to complete charting with the provided script. For accuracy, we counted the number of charting errors made while reading the script. We classified three different types of errors: (1) repeat error—when the system did not respond and the participant had to repeat the command, (2) wrong response error—when the system's response differed from the participant's input, and (3) insertion error—when the system charted something, but the participant had not said anything. In order to analyze the impact of the errors, we further characterized misrecognitions and insertion errors into two different categories: disruptive error: a system's response that modified the chart and caused the participant to be off script—e.g., the participant

said “existing” and the chart closed, “exit”; and non-disruptive error: a system response that was incorrect but was not off script—for example, when the participant said “tooth 3” and the system selected “tooth 30.” To evaluative user satisfaction, we manually reviewed the answers to the open-ended user-satisfaction questionnaires. We then classified the answers into categories and totaled them.

We performed a statistical analysis to determine whether any system performed better than the other systems. First, we used SPSS to perform the Kolmogorov-Smirnov and Shapiro-Wilk tests—both tests for normality [89]—and determined that none of our data were normally distributed. Therefore, to compare the average number of errors and amount of charting time among the four systems, we used SPSS to perform the non-parametric Friedman test. We chose the Friedman test because for each calculation we have one dependant variable (time or error) and we have one within-subjects independent variable (the system being used) with two or more levels (each of the four systems). For this test we chose a significance level of 0.01.

HARDWARE. All PMSs were installed in their default configurations on a Windows XP, 1.5GHz Intel Pentium 4 processor, 256MB of RAM computer equipped with an extra 80GB hard drive. The extra hard drive stored images of each software package installed in the default configuration. Fresh software installation images were used for each test eliminating any influences from other users’ speech profiles or other users’ task results.

4.2.2 Results

Two participants did not complete sessions for all four system tests. We discarded their data, which left us with 18 students and a total of 18 tests of each of the four systems. Fourteen of the students were male and four female and all spoke American English as their native language.

Four of the students were in their first year of dental school, 11 in their second year, two in their third year, and one in his fourth year. From the background questionnaire [87], we learned that participants spent an average of 9.8 hours a week using a computer for personal use compared to the 4.8 average hours per week for professional use. Participants reported using a computer for mainly email and Internet browsing. Eighty-three percent of the students reported being self-taught to use computers, 67 percent had college courses in computer use, and only 17 percent had computer classes in dental school. Finally, the majority of the students (89%) reported themselves either sophisticated or very sophisticated computer users. The shortest time between session visits was two weeks, however this only occurred with one participant. Most of the participants scheduled sessions four or more weeks apart.

TIME. Average time to chart using the script was broken down into two categories: hard tissue and periodontal charting. The hard tissue results include data from Dentrix and EagleSoft—the only two systems that allowed hard tissue charting via speech. The periodontal results include data from all four systems. The average time to complete the speech training for all of the systems was 6:52 (minutes: seconds). Table 4 shows the average time for training and to chart hard tissue and periodontal findings using the script for each system.

Hard tissue. The average time to chart the nine hard tissue findings was 2:48, Dentrix took an average of 2:58 and EagleSoft an average of 2:37—there were no significant differences between the times to chart the hard tissue findings in these two systems.

Periodontal. The average time to chart the nine periodontal findings was 2:06. EagleSoft took the least time, with an average of 1:49 and PracticeWorks took the longest taking an average 2:17—there were no significant differences among times to chart periodontal findings.

Total exam. The results for total exam include only Dentrix and EagleSoft data and include the time it took to select a patient via speech. The average time to select a patient and chart all 18 findings was 5:02. Dentrix took an average of 5:31 while EagleSoft took an average of 4:32—there were no significant differences between the times to chart the entire exam in these two systems.

Table 4. Average times in seconds. Reported times for charting exams include time to repeat words/phrases. Total exam times include time to select the patient via speech. Lower and upper 95% confidence intervals (CI) appear in parenthesis.

	training		hard tissue (95% CI)	perio (95% CI)	total exam (95% CI)
Dentrix	518.7		177.9 (164.5, 191.2)	136.3 (41.0, 231.5)	331.4 (231.9, 431.0)
EagleSoft	315.2		157.3 (144.7, 170.0)	109.4 (99.0, 119.8)	272.2 (253.1, 291.4)
PracticeWorks	421.9		-	137.4 (118.0, 156.8)	-
SoftDent	392.5		-	119.4 (100.2, 138.5)	-
average totals	412.1		167.6	125.6	301.8

When broken down into time required to chart each finding, the average time to chart a finding in the hard tissue exams (Dentrix and EagleSoft) was 17.9 seconds. Average time to chart a finding in the periodontal exams (all four systems) was 17.1 seconds. Figure 8 shows the average times to chart each finding with each system. In Figure 8, charting *furcation on tooth three* appears to have taken the longest amount of time, but that measurement was actually a combination of charting three separate findings: a buccal furcation of two, a distal furcation of one, and a mesial furcation of one. We grouped together the three findings and timed them as one finding, because charting furcation in SoftDent required several mouse clicks and it was infeasible to time each furcation finding separately.

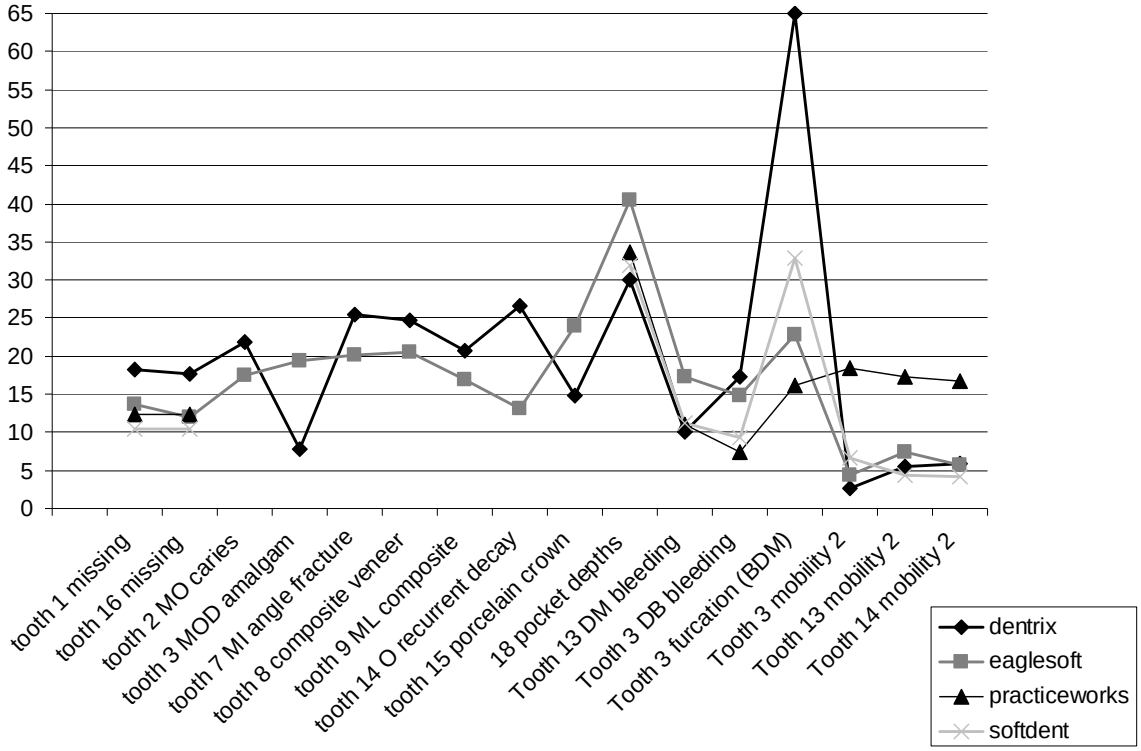

Figure 8. Average time to chart each finding with each system. *Tooth 3 furcation* is the sum of three findings. B-buccal, D-distal, M-mesial, O-occlusal, I-incisal, L-lingual.

ERRORS. There were a total of 350 repeat errors, 139 misrecognition errors, and 53 insertion errors in all exams. As Table 5 shows, the most frequently occurring error was a repeat error (average of 4.9 per exam). The least occurring error was an insertion error (average of 0.7 per exam). EagleSoft had the highest number of errors per exam (12.2) and SoftDent had the smallest number of errors per exam (0.4).

Table 5. Average number of errors per exam (n= total number). There were 18 exams per system for a total of 72 exams.

	insertion	misrecognition	repeat	all errors
Dentrix	2.2 (n=40)	3.1 (n=55)	4.4 (n=79)	9.7 (n=174)
EagleSoft	1.0 (n=10)	1.6 (n=28)	10.0 (n=181)	12.2 (n=219)
PracticeWorks	0.1 (n=2)	2.8 (n=51)	4.9 (n=89)	7.9 (n=142)
SoftDent	0.1 (n=1)	0.3 (n=5)	0.1 (n=1)	0.4 (n=7)
average (ttl)	0.7(n=53)	1.9 (n=139)	4.9 (n=350)	7.5 (n=542)

When we analyze the repeat and misrecognition errors, the most frequently repeated word was "ok." repeated 33 times (out of 347 total repeat errors). The most misrecognized phrase was "three, two, one." (pocket depths for a tooth). The list of the ten most repeated and most misrecognized words can be seen in Table 6.

Table 6. Ten most repeated and misrecognized words/phrases.

#	repeated words/phrases	misrecognized words/phrases
1	"ok" (n=33)	"three, two, one" (n=26)
2	"mesial" (n=23)	"select three" (n=10)
3	"one" (n=20)	"select fourteen" (n=10)
4	"two" (n=20)	"bleeding on tooth thirteen distal mesial" (n=8)
5	"select thirteen" (n=18)	"select thirteen" (n=7)
6	"select fourteen" (n=17)	"ok" (n=7)
7	"bleeding distal buccal" (n=13)	"bleeding distal buccal" (n=7)
8	"select three" (n=12)	"tooth three" (n=5)
9	"cancel" (n=9)	"tooth thirteen" (n=5)
10	"lingual" (n=8)	"three" (n=5)

There were 178 non-disruptive errors in all the sessions—an average of 9.9 per exam. There were only 15 disruptive errors (errors resulting in the system going off script) in all of the sessions—an average of 0.8 per exam. All of the disruptive errors were due to misrecognitions—the most commonly misrecognized phrase that resulted in a disruptive error was "select fourteen." This phrase was misrecognized four times (three in Dentrix and once in PracticeWorks).

Table 7. Average significance of errors per exam (n= total number). Disruptive and non-disruptive errors include only misrecognitions and insertions. There were 18 exams per system for a total of 72 exams.

	disruptive	non-disruptive
Dentrix	0.4 (n=7)	4.9 (n=88)
EagleSoft	0.3 (n=5)	1.8 (n=33)
PracticeWorks	0.2 (n=3)	2.8 (n=51)
SoftDent	0 (n=0)	0.3 (n=6)
totals	0.8 (n=15)	9.9 (n=178)

Hard tissue. When charting the nine hard tissue findings Dentrix had a total of 94 errors (49 repeat, 26 insertion, and 19 misrecognition) and EagleSoft had a total of 124 errors (113

repeat, 2 insertions, and 9 misrecognitions) —there were no significant differences between the number of errors that occurred during hard tissue charting in these two systems.

Periodontal. Table 8 shows the number of errors for each system while charting the nine periodontal findings. PracticeWorks had the highest number of errors with 142 and SoftDent had the smallest number, seven. SoftDent had significantly fewer errors than the three other systems: Dentrix $X^2 = 14.2$, $d.f. = 1.0$, p=0.000; EagleSoft $X^2 = 17.0$, $d.f. = 1.0$, p=0.000; PracticeWorks $X^2 = 18.0$, $d.f. = 1.0$, $p = 0.000$. PracticeWorks had significantly more errors than both SoftDent ($X^2 = 18.0$, $d.f. = 1.0$, $p = 0.000$) and Dentrix ($X^2 = 12.25$, $d.f. = 1.0$, $p = 0.000$) —there were no other significant differences.

Table 8. Number of errors while charting the **periodontal** exams. There were 18 exams per system for a total of 72 exams. Numbers in parenthesis is percent of error based on total number of periodontal speech commands in all 18 exams for each system.

	insertion	misrecognition	repeat	total
Dentrix	14 (3.2%)	36 (8.7%)	30 (7.2%)	80 (19.3%)
EagleSoft	8 (1.5%)	19 (3.5%)	68 (12.6%)	95 (17.6%)
PracticeWorks	2 (0.6%)	52 (15.2%)	88 (25.7%)	142 (41.5%)
SoftDent	1 (0.2%)	5 (1.2%)	1 (0.2%)	7 (1.6%)
total	25	112	187	324

USER SATISFACTION. Table 9 shows the categorized responses to the satisfaction questionnaires. Users favored different things about each system. Accuracy was listed as the most favored aspect of SoftDent (52%), usability the most favored for both Dentrix (42%) and PracticeWorks (28%) and valuable feedback the most favored for EagleSoft (48%). More common trends appeared in users' opinions on the negative aspects of systems. Having to repeat commands was listed as the least favored aspect of Dentrix (38%), EagleSoft (39%) and PracticeWorks (55%). Usability was the least favored aspect of SoftDent (35%) and Dentrix (38%). In all systems but PracticeWorks, the majority of the users' experience with the system matched their expectations. In all four systems, most users only wanted to repeat a word once

during an exam. Users were clearly willing to enunciate their speech more clearly to use a system like these in practice. Finally, in all systems but PracticeWorks, a majority of the students would use the system in clinical practice.

Table 9. Responses to the satisfaction questionnaire. Q1 & Q2 can have more than one response. Q3-Q6 n= 18. Calculation errors due to rounding.

	responses	Dentrix	EagleSoft	PracticeWorks	SoftDent
Q1. good things about system	**accurate**	21%(4)	14%(3)	22%(4)	52%(12)
	usable	42%(8)	29%(6)	28%(5)	35%(8)
	feedback	32%(6)	48%(10)	11%(2)	0
	efficient	5%(1)	5%(1)	17%(3)	13%(3)
	training	0	5%(1)	22%(4)	0
Q2. bad things about system	**repeat**	38%(6)	39%(9)	55%(12)	6%(1)
	error	0	26%(6)	27%(6)	12%(2)
	slow	19%(3)	22%(5)	5%(1)	6%(1)
	use mouse	0	4%(1)	0	18%(3)
	usability	38%(6)	9%(2)	9%(2)	35%(6)
	feedback	6%(1)	0	5%(1)	24%(4)
Q3. match your expectations	**yes**	56%(10)	72%(13)	28%(5)	89%(16)
	no	44%(8)	28%(5)	72%(13)	11%(2)
Q4. acceptable repetition frequency	**1 per exam**	44%(8)	56%(10)	50%(9)	50%(9)
	2 per exam	17%(3)	17%(3)	28%(5)	11%(2)
	≥ 3 per exam	39%(7)	28%(5)	22%(4)	39%(7)
Q5. willing to enunciate your speech more clearly	**yes**	83%(15)	94%(17)	83%(15)	89%(16)
	no	6%(1)	0	6%(1)	0
	maybe	11%(2)	6%(1)	11%(2)	11%(2)
Q6. enter clinical data faster with system	**yes**	56%(10)	67%(12)	39%(7)	72%(13)
	no	22%(4)	17%(3)	22%(4)	6%(1)
	maybe	22%(4)	17%(3)	39%(7)	22%(4)

Based on our qualitative questionnaire it would not be feasible to perform a statistical analysis to demonstrate correlations with user satisfaction and system performance. However, some patterns did emerge. When identifying the positive aspects of a system, 52 percent of the students noted SoftDent's ability to chart correctly, i.e., its accuracy (see Table 9). This finding was not surprising considering SoftDent was the system with the fewest errors. Further, when identifying negative aspects of a system, only 18 percent of students noted the number of errors that occurred or the need to repeat words in SoftDent. Again, this may be attributed to the fact

that SoftDent was the system with the fewest number of errors. When identifying negative aspects of each system, 38% or more of the students commented on the need to repeat words—the most common of all error types—in Dentrix, EagleSoft, and PracticeWorks. PracticeWorks—the system with the highest number of errors—was the only system for which the majority of user experiences did not match their expectations and for which the majority of the participants either answered "no" or "maybe" when asked if they believed they could enter clinical data faster with this system.

4.2.3 Discussion

Dentists have made efforts to adopt the speech interfaces of existing practice management systems [7]. Nonetheless, as this study and its preliminary work [8] show, existing systems require tedious step-by-step speech commands, and dentists could have to repeat as many as ten commands per exam depending on which system they were using. The time needed to complete the exam was relatively short, with an average of five minutes to select a patient and chart all 18 findings. However, participants used a prepared script to chart. SoftDent—the system where charting had the fewest errors and took the shortest amount of time—was the system with the smallest amount of the exam able to be charted via speech. SoftDent only allowed periodontal charting via speech and only 73 percent of the nine periodontal findings could be completed via speech alone. When looking at Dentrix and EagleSoft—the only two systems that allow charting of both hard tissue and periodontal findings—charting was faster in EagleSoft, but Dentrix makes fewer errors. There were no significant differences between these two systems. The scripts also show that using a keyboard and/or mouse was necessary in EagleSoft, PracticeWorks and SoftDent, which defeats the purpose of using speech recognition as a hands-free way to

interact with the computer. If a clinician were to use one of these systems, she would still have to de-glove to interact with the computer and the keyboard and mouse would still need to be easily accessible.

User satisfaction seemed to correlate with the time and error findings—however, based on our qualitative questionnaire it would not be feasible to perform a statistical analysis to show this. Accuracy was reported as the highest positive aspect of SoftDent—the system with the fewest errors—while the need to repeat words and number of errors were ranked the lowest of the negative aspects The opposite was true for PracticeWorks—the system with the highest number of errors—for which the need to repeat words and the number of errors were the highest ranked negative aspect. One telling finding from the general comments in the user satisfaction questionnaires is the multiple times participants made statements—for example, “I didn’t like not being able to say what I want” or “Remembering the phrases to use to activate different features of the program may be difficult.” These statements identify the desire for a natural speech interaction with the systems.

Across all four systems, the most common frustration a participant had to deal with was the system not responding to a command and thus needing to repeat a word. There are many factors that could cause repeat errors. It was possible that the word spoken by the participant was simply misrecognized and because the system did not understand the misrecognized word, it did not respond. Also, across all of the systems, the word “three” or some variation of it (e.g., thirteen) appears in more than half of the ten most misrecognized words. Misrecognizing “three” may be due to the voiceless fricative <th> appearing to be "noise" on a spectrograph—as such, when the speech recognition engine tries to control for background noise, it actually removes some of the voiceless fricative pronunciation [90].

Through our analysis we are able to show that SoftDent had significantly fewer errors than all other systems and PracticeWorks had a significantly higher number of errors than two of the systems when charting the periodontal findings. This is interesting considering both software packages are produced by Kodak (Atlanta, GA)—one reason for this may be training. PracticeWorks was the only system that did not use the traditional Microsoft Speech training. Instead of reading a story, PracticeWorks asked the user to read lists of numbers and dental terms. It is possible that reading words out of a context—as in the form of lists—causes the user to read with different tones and inflections and thus recognition during the exam could be reduced.

This study is the first step in achieving our long-term research goal of developing a natural language interface that will allow clinicians to naturally speak as a means of entering data in a dental computer-based patient record. Existing systems do not allow dentists to complete hard tissue and periodontal charting via naturally spoken text. The absence of a flexible, robust, and accurate natural language interface is a significant barrier to the direct use of computer-based patient records by dental clinicians. Because of poor user interface and speech application design, data entry is still being conducted using the assistant as a "remote control." This situation reduces the potential benefit of chairside clinical decision support systems—which are most effective when used directly by the decision-maker—and prevents highly-trained auxiliary personnel from performing value-added tasks while engaged as computer operators. In the second objective of this dissertation we begin the development of a better and more natural speech interface by created and evaluating a speech-to-chart prototype for charting naturally-spoken dental exams.

5.0 OBJECTIVE 2: CREATE & EVALUATE SPEECH-TO-CHART PROTOTYPE

The results of the evaluation of the existing practice management systems indicated a clear need for a natural-language speech interface for dental charting. Thus, our research team was motivated to develop and evaluate a speech-to-chart prototype for charting naturally spoken dental exams. For this objective, we built a prototype speech-to-chart pipeline that could not be implemented in real-time but simulates a real-time system. We used resources that are currently available and that we have developed specifically for this project. In this section, we evaluated how well speech recognition works on dictated dental exams; evaluated how well the exams can be interpreted and structured by a natural language processing (NLP) application and; investigated the impact of speech recognition errors on NLP performance on a dental charting task.

5.1 METHODS

We developed and evaluated a speech-to-chart prototype for charting naturally-spoken hard tissue dental exams. The system is made up of the following components—as shown in Figure 9): (A) a speech recognizer, which creates a transcript of the exam; (B) a speech recognition post-processor for error correction; (C) an NLP application (ONYX) to extract and structure the

information from the text; and (D) a graphical chart generator for charting the NLP output on a commercial electronic dental record system.

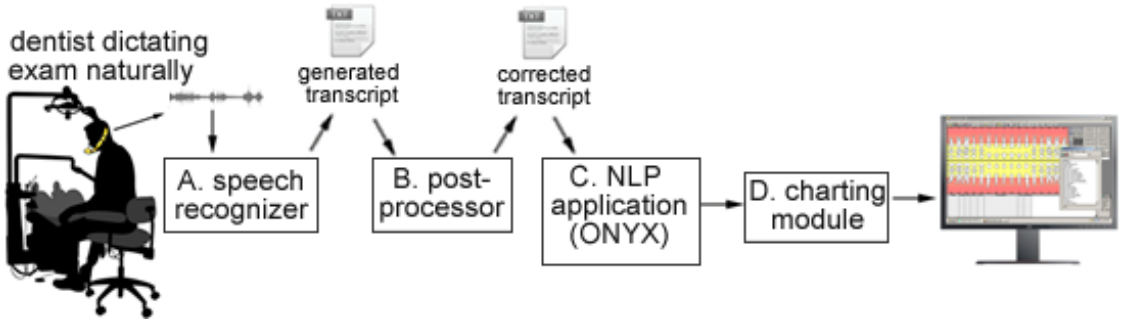

Figure 9. Components of speech-to-chart prototype.

Our speech-to-chart prototype uses commercially available software for speech recognition and for charting findings. We developed a speech post-processing algorithm and the NLP system for interpreting the speech transcript. The following sections describe the development and preliminary evaluations of each component of the system, followed by a summative evaluation of the entire speech-to-chart prototype.

5.1.1 Datasets

We collected audio digital recordings of initial hard tissue exams from six dental students from the University of Pittsburgh School of Dental Medicine. We gave each student a digital recorder (M-Audio MicroTrack II Professional 2-Channel Mobile Digital Recorder, M-Audio USA, Irwindale, CA) to record dictations of initial hard tissue exams. We instructed the students to make a recording while examining a patient as if they were dictating to auxiliary personnel. The students wore masks and used a clip-on microphone while recording exams. We were not present for any of the recordings and we did not enforce our instructions. After listening to some of the

recordings, a dentist from our research team felt that some of the recordings were dictated using the chart after the student examined the patient. To ensure high quality audio recordings, the digital recording device had a 16 Bit, 44 kHz sampling rate and was supplemented with a two gigabyte compact flash memory card. The Institutional Review Board at the University of Pittsburgh reviewed and approved this research protocol (IRB# 09090022).

From the six students we collected 25 recorded exams that we divided into separate development and test datasets. The Development Set consisted of 13 exams from four of the students. We used the Development Set to evaluate the speech recognition accuracy and to develop our post-processing error-correction algorithm. The Test Set comprised 12 exams from all six dental students. We used the Test Set to evaluate speech recognition accuracy with and without the post-processing error correction algorithm and to evaluate the speech-to-chart prototype.

5.1.2 Components of the speech-to-chart prototype

A. SPEECH RECOGNIZER. The first task of the speech-to-chart prototype was to generate text from the audio dictation of the exam. To do this we used an out-of-the-box transcription software program called Dragon Naturally Speaking 9.0 (Nuance, Burlington, MA). Dragon NaturallySpeaking is one of the leading transcription software on the market [91]. One reason for Dragon's success is that a reasonable amount of ambient noise has no effect on Dragon's transcription accuracy [11]. Dragon requires users to build a voice profile so that it can learn the user's voice and speech patterns. We used an excerpt from "Alice in Wonderland," one of Dragon's training stories, to create an individual voice profile for each student. All transcriptions of dictated dental exams were generated using each student's individual profile. In this study,

each student recorded at least one exam where the volume was too soft for Dragon to transcribe and for two students the "Alice in Wonderland" excerpt contained too much noise. Therefore, before feeding exams to Dragon, we processed every exam with an open-source audio recording and editing software program—Audacity v.1.2.6 [92]—to enhance the recording's volume and remove noise (if necessary). We created two transcripts for each exam in the Development and Test Sets: the first was generated by Dragon; the second was manually transcribed by a medical transcriptionist. We used the manually transcribed exam to evaluate speech recognition accuracy.

We evaluated Dragon's accuracy in transcribing dental exams using SCLITE [93] to calculate the percent word accuracy of the transcriptions. SCLITE—a tool developed by the US National Institutes of Standards and Technology for scoring and evaluating the output of speech recognition systems [93] —uses the Levenshtein distance to align the software transcription with the human transcription identifying correct words, insertions, deletions, and substitutions. After the texts have been aligned, percent word accuracy is calculated as:

$$\left[\frac{\text{\# of correct words}}{\text{\# of reference words from the human transcript}}\right] * 100$$

In addition to assessing the accuracy of speech recognition transcription on dental exams, we performed an error analysis on Dragon's output to characterize the nature of the errors. We labeled each error with an error type. For example "mesial" was misrecognized as "me feel," and we labeled the error as a "sounds like" error. Table 10 describes each error type classification.

Table 10. Description of Dragon error classifications.

classification	description	example
sounds like	entire word or phrase sounds like correct word	"me feel" for "mesial"
starts with	the beginning of the word or phrase sounds like the correct word	"too" for "tooth"
ends with	the end of the word or phrase sounds like the correct word	"amazing" for "missing"
homophone	word with the same sound but different spelling as correct word	"to" for "two"
spelling	incorrect spelling of correct word	"carries" for "caries"
acronym	acronym for incorrect word	"DK" for "decay"
number	misrecognized number	"30^{th}" for "30"
other	all other unclassified errors	"the" for "with"

B. POST-PROCESSING ERROR CORRECTION ALGORITHM. We did not expect perfect speech recognition performance in this new domain of dentistry. Common approaches to improving speech recognition include adapting acoustic models (probabilities of speech sounds), grammars (probability of a word given a history of previous words), and pronunciation dictionaries (pronunciations of words using one or more phonetic transcriptions) to the new domain [94-98]. However, these methods are beyond the scope of this dissertation. To address issues with recognition accuracy in this project, we created a post-processing algorithm for automated error-correction of transcripts.

An advantage to using post-processing error-correction over directly improving the speech recognition application is that post-processing is software independent and is not tied to a particular recognition system. Post-processing error-correction accounts for the fact that many of the best speech recognizers are privately owned and thus restrict the ability to manipulate and improve the application [99]. Also, post-processing provides the ability to fix errors that were mistakenly introduced by the speaker [99]. Some common post recognition error-detection methods include co-occurrence, edge-based similarity, and pattern matching [99].

Co-occurrence algorithms use statistical methods to determine the number of times a word occurs in a specific context [99]. With a large enough training corpus of relevant texts, one can calculate co-occurrence relations and use them to determine the probability of a word

occurring in the context of other words, thus allowing the identification of possible misrecognitions. Unfortunately—unlike other medical domains—dentistry is still adopting computerized medical records and therefore a large corpus of dental exams is not available.

Edge-based similarity is a method that involves utilizing a conceptual distance metric within the Unified Medical Language System (UMLS) —a comprehensive ontology of medical language [99]. In the UMLS, similar items or concepts appear near each other [99]. Using edge-based similarity, one can find errors in transcripts by referencing the UMLS to calculate the semantic distances between two concepts to determine if the concepts should appear together. While building our semantic model for ONYX (Section 2.7), we explored the UMLS and because of its limited coverage of dentistry, we found that dental concepts we identified in our texts did not map well to UMLS concepts. Because the UMLS has limited dental entries and the dental domain does not have a large corpus of electronic dental exams, co-occurrence and edge-based similarity algorithms are not ideal for post-processing of dental transcriptions. Therefore, we explored the non-statistical rule-based method of pattern matching.

Pattern matching involves creating a database of common error patterns in a particular domain and then using rules to identify instances of those patterns in the speech recognition transcripts [99]. Our post-processing algorithm is based on this method. We analyzed the training data in the Development Set to identify common error patterns in dental exams. We then created a rule-based algorithm to correct errors found in the transcriptions of the Test Set. We do recognize that while pattern matching can be very accurate in identifying errors it also has limitations: error patterns that do not appear in training set and thus do not have corresponding values in the database can not be identified, and the method is susceptible to false positives when correct words happen to occur in an identified error pattern [99].

Using the data from the error analysis we conducted on Development Set exams transcribed by Dragon, we created a post-processing algorithm for correcting Dragon's common errors. The algorithm involves simple substitution of erroneous transcribed words with the correct words and performs three types of corrections—the code for the algorithm can be found in Appendix B. The first technique corrects spelling and homophone errors—for example, "carries" is replaced with "caries." The second technique addresses errors in which a single word is transcribed as multiple words—for example, the words "amount of" are replaced by the single word "amalgam." The third is a context-sensitive spelling and homophone correction that substitutes words that appear in a particular context—for example, "to" is replaced with "two" when preceded by the word "number."

To evaluate the post-processor, we measured change in percent word accuracy before and after post-processing.

C. NLP APPLICATION (ONYX). We developed an NLP application called ONYX for extracting relevant information from the transcripts and organizing the information for automated charting [41]. ONYX implements syntactic and semantic analysis to interpret sentences using a combination of probabilistic classifiers, graphical unification, and semantically annotated grammar rules [41]. From a sentence in the exam, ONYX fills in templates for the following types of information: dental conditions (caries or fractures); restorations (fillings and crowns); tooth locations and; modifiers (tooth part and tooth surface) [41]. ONYX then fills in templates of words and concepts and attempts to identify relationships between concepts [41]. ONYX's output after parsing an exam is a list of predicate logic statements that identify chartable conditions [41]. For example, for the sentence "There is a cavity on tooth 2" ONYX predicate statement output is:

CONDITIONAT(*CARIES, *NUMBERTWO) & LOCATIONHASSURFACE(*NUMBERTWO, *MESIAL).

Text generated by Dragon does not contain punctuation—however, ONYX requires punctuation to segment sentences and process exams. Therefore, we manually entered punctuation into each transcription.

D. GRAPHICAL CHART GENERATOR. After we processed each exam with ONYX, we used regular expressions to parse ONYX's output and chart the exam in a commercial charting system. We developed a Python program that extracts conditions, restorations, surfaces, and other information necessary for charting. Each concept that was outputted by ONYX was processed individually, and the extracted information was passed to functions that manipulate the Microsoft Windows operating system (Microsoft, Richmond, VA) to interact with the Dentrix v.10 (Henry Schein, Melville, NY) charting system. Each finding for each tooth was ultimately charted in Dentrix—a leading practice management software system that allows electronic charting of patient hard tissue exams [100].

5.1.3 Summative evaluations

We evaluated three versions of the prototype charting application. Each version processed ONYX's output to create a chart in Dentrix but used different transcriptions of the dental exam. The first version used a transcript created by a human transcriptionist listening to the exam; the second used a transcript generated by Dragon Naturally Speaking after training on the dental student dictating the exam; the third version used a transcript that was generated by Dragon but corrected with our simple post-processing algorithm. We compared the output of each version against electronic gold standard charts—see Figure 10. The gold standard consisted of an electronic chart that was manually entered by a dentist directly from the reference transcripts. A

second dentist reviewed the gold standard chart for verification. All inconsistencies found by the second dentist were discussed with the author and a consensus decision on the correctness of the charted information was reached. The summative evaluation addresses the following questions: How accurate are charts generated by our NLP system from perfect speech transcriptions? And how much does our speech-to-chart system's performance degrade with automatically-generated speech transcripts?

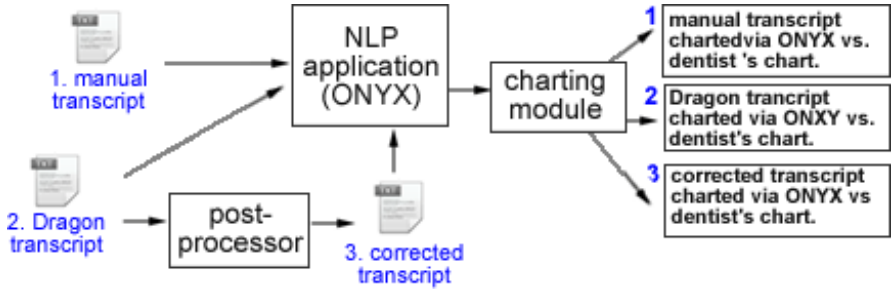

Figure 10. Summative evaluations of speech-to-chart prototype.

To answer these questions, we manually reviewed each chart and compared it against the gold standard chart. A true positive was identified if the student stated a finding was present and the automatically-generated output matched that finding exactly—including finding, tooth number, surface, and necessary modifiers. We also labeled an item as a true positive if the student stated a tooth was "fine" and the automatically-generated output identified the condition of that tooth as "normal." A true negative was identified when the student stated nothing about a tooth and the system did not chart a finding for the tooth. A false positive was identified when the system charted a finding on a tooth when no finding was stated. Finally, we identified an item as a false negative when the system did not chart or incompletely charted a finding that was present in the gold standard chart—for example, tooth number and finding were correct but the surface was incorrect. Defining true negatives can be complex as there are hundreds of possible

statements concerning the findings for a tooth that are not mentioned in an exam. Likewise, false negatives are difficult to define due to partially correct findings. When the system identified the correct tooth number and finding but incorrectly identified a surface, it was able to get the majority of the finding correct. For this study, we chose to define true negatives and false negatives according to the perspective of the dentist: when any part of a finding was wrong on the chart, the finding was considered incorrect—a false negative—because a dentist would ultimately have to correct that finding. From our manual comparison, we calculated the following outcome measures:

$$\text{accuracy} - \frac{(TP + TN)}{\#\ findings + allpossibleTN} \qquad \text{positive predictive value} - \frac{TP}{TP + FP}$$

$$\text{sensitivity} - \frac{TP}{TP + FN} \qquad \text{negative predictive value} - \frac{TN}{TN + FN}$$

$$\text{specificity} - \frac{TN}{TN + FP}$$

We performed a statistical analysis to compare accuracies for each exam type. Since our data is normally distributed, we used SPSS to perform the analysis of variance one-way Anova test. We chose the one-way Anova because we have one dependant variable (accuracy) and we have one independent variable (exam type) with two or more levels. For this test we chose a significance level of 0.05. Finally, we conducted an error analysis of the automatically-charted exams ONYX generated from the manual transcriptions. Performing the analysis on the manually transcribed (i.e. perfect) exams allowed us to identify errors made by our natural language processing system, ONYX, and not by the speech recognizer.

5.2 RESULTS

5.2.1 Datasets

All of the participants who provided initial hard tissue exams for this study were male, American-English speaking dental students at the University of Pittsburgh's School of Dental Medicine. The Development Set—made up of 13 exams from four students—had a total of 345 findings for an average of 27 findings per exam. Approximately 66 percent of the findings were conditions (66% of those caries) and 28 percent were restorations (57% of those were amalgams and 10% were crowns), the final six percent where instances when the participant said the tooth was normal (e.g., "number 2 fine."). The Test Set—made up of 12 exams from all six students—had a total of 340 findings for an average of 28 findings per exam. Approximately 62 percent of the findings were conditions (50% of those were caries) and 31 percent were restorations (48% of those were amalgams and 25% were crowns), the final seven percent where instances when the participant said the tooth was normal.

While creating the gold standard exams for the summative evaluation, the second dentist corrected 13 findings from the original dentist's exams.

5.2.2 Transcribing exams

Error analysis of Dragon's output. Dragon's average percent word accuracy for the 13 exams in the Development set was 63.7 percent. Dragon's accuracy was 11.3 percent higher for non-dental sentences (Table 11), but non-dental sentences only contained four percent of all words in the

exams. Dragon's average percent word accuracy for the 12 exams in the Test set was 63.5 percent.

Table 11. Percent word accuracy and types of errors calculated via SCLITE (Development Set).

sentences	% word accuracy	words	errors	errors per/exam	% substitutions	% insertions	% deletions
all	63.7%	2392	955	73	70	18	12
non-dental	75.0%	90	46	3	74	11	15
dental	63.5%	2264	909	69	70	18	11

The results for the error analysis on the Development Set can be found in Table 12—we did not perform an error analysis on the Test Set. Dragon misrecognized a total of 661 terms out of 2392 words—the largest percent (34%) of errors were "sounds like" errors. Surfaces were the semantic type most often misrecognized.

Table 12. Types of errors manually identified not including deletions & insertion (Development Set). Percent total errors due to rounding.

	sounds like	starts with	ends with	homo-phone	spelling	other	acronym	number	total
all sentences	227 (34%)	80 (12%)	65 (10%)	25 (4%)	97 (15%)	85 (13%)	19 (3%)	63 (10%)	661
non-dental	5 (18%)	1 (4%)	3 (11%)	1 (4%)	0	11 (39%)	0	7 (25%)	28
dental	222 (35%)	79 (12%)	62 (10%)	24 (4%)	97 (15%)	74 (12%)	19 (3%)	56 (9%)	633
dental: surface	62 (38%)	29 (18%)	27 (17%)	1 (0.6%)	22 (14%)	17 (10%)	4 (2%)	-	162
dental: condition	23 (18%)	7 (5%)	4 (3%)	-	75 (58%)	7 (5%)	14 (11%)	-	130
dental: restoration	43 (75%)	5 (9%)	8 (14%)	-	-	1 (2%)	-	-	57
dental: tooth number	1 (1%)	-	-	21 (28%)	-	-	-	54 (71%)	76
dental: anatomic location	37 (59%)	22 (35%)	-	-	-	3 (5%)	-	1 (2%)	63

5.2.3 Post-processing error correction algorithm

We created a post-processing algorithm containing three techniques for correcting Dragon's common errors. The spelling and homophone-correction technique corrected two common

errors. The N for M technique—in which a single word is transcribed as multiple words—identified 78 commonly misrecognized words or phrases and their correct counterparts. The longest phrase contained three words "can pause it" and was corrected as "composite." The context-sensitive spelling and homophone correction—or n-gram—technique contained three routines. The first routine corrected common errors with the word "to." Based on previous or following words, the word "to" would be changed to "tooth" or "two." Next, based on previous or following words, the word "for" could be changed to "four." Finally, Dragon commonly misrecognized the number "10" as the word "and." Again, based on previous or following words the algorithm would correct the word.

Percent word accuracy increased after each post-processing technique (see Figure 11) with a 3 to 14 percent increase in accuracy after applying all three techniques—56.0 to 70.7 percent on the Development Set and from 63.5 to 66.6 percent on the Test Set. The post processing algorithm made 320 changes to the Development Set and 111 changes to the Test Set.

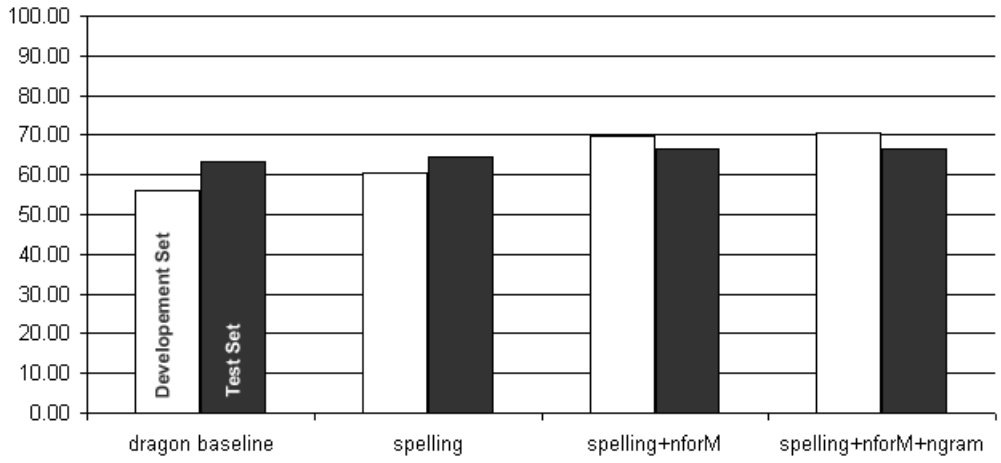

Figure 11. Percent word accuracy for each algorithm technique.

Table 13. Changes made by each post-processing algorithm technique.

algorithm technique	Development Set	Test Set
spelling	97	55
NforM	188	45
n-gram	35	11
all combined	320	111

5.2.4 Summative evaluations

Table 14 shows average performance for each version of the prototype charting application: Dragon transcripts, Dragon with post-processing routines, and manually-transcribed transcripts. Using manual transcripts, ONYX's accuracy ranged from 53 to 100 percent across exams, with a mean accuracy of 80 percent. Sensitivity averaged 75 percent, and positive predictive value was high at 92 percent, resulting from only 27 false positives. Performance degraded when using speech-generated transcripts, with average accuracy dropping from 80 to 48 percent with Dragon—however, post-processing the exams increased accuracy from 48 to 54 percent.

Table 14. Performance of end-to-end charting system using Dragon transcripts (D), Dragon with post-processing routines (D+PP), and manually transcribed transcripts (MT).

transcription	D	D+PP	MT
accuracy	48	54	80
sensitivity	31	43	75
specificity	68	73	74
positive predicative value	89	87	92
negative predictive value	30	37	49

Accuracies differed significantly across the three transcript types, F (2, 33) = 7.19, p = 0.003. Tukey post-hoc comparisons of the three transcript types indicate that the manual transcripts (M = 0.798) had significantly higher accuracies than both the Dragon plus post-processing exams (M = 0.541), p = 0.019 and the Dragon transcripts (M = 0.476), p = 0.003. Comparisons between the Dragon transcript (M = 0.476) and the Dragon plus post-processing transcripts (M = 0.541) were not statistically significant at $p < 0.05$.

Performance varied widely based on the student dictating the exam. For the manually transcribed data, average accuracy per student ranged from 99 to 54 percent, with the average accuracy per student being 74 percent.

5.2.5 Error analysis of manually transcribed exams

When using the manually-transcribed exams, ONYX made two types of errors: (1) false negatives—when it missed charting a finding or an aspect of a finding and (2) false positives—when it added a finding that was not dictated in the exam. We classified ONYX's false negative errors into three categories: (1) errors with surfaces, (2) errors due to lack of training cases, and (3) miscellaneous errors. Seventeen percent of the false negatives were misidentified surfaces. In all of the surface errors, the system identified the correct condition (i.e., caries) or restoration (i.e., amalgam) but incorrectly identified the surface. In some cases, a finding would have more than three surfaces and the system would only identify one or two of them. Other times a surface was stated in a way the system has never been trained on (i.e., "occluso" for "occlusal"). The second type of error was due to lack of training cases, which happened with 48 percent of the false negatives. Many times a finding was not charted because the system had either never been trained on that type of finding (i.e., periapical abscess) or a finding was expressed in new way (i.e., "extracted" instead of "missing"). Finally, like any automatic system there were miscellaneous errors, which accounted for the final 35 percent of the false negatives. Some of these errors occurred when two findings were stated in one sentence—for example "also on the mandible teeth 18 and 19 are both pfms". Miscellaneous errors also occurred with the misinterpretation of a finding—for example, "6 is a disto lingual composite, which is fine" was just interpreted as "tooth 6 fine" and the composite was missed.

ONYX only charted 27 false positives in all 12 exams. False positives occurred for mainly two reasons: (1) ONYX interpreted a treatment plan as a finding, and (2) ONYX incorrectly identified a tooth number. ONYX currently does not have a sophisticated discourse processor, therefore when it found sentences like "number 2 has an occlusal amalgam with mesial and distal decay, so 2 will be an MOD amalgam", ONYX charted the occlusal amalgam, the mesial and distal caries, and the MOD amalgam. The MOD amalgam which was part of the treatment plan resulted in a false positive. Next, in some instances ONYX incorrectly identified a tooth number. ONYX's discourse processor is designed to use the previous tooth number if one is not provided. This was designed for situations like "number 9 has a distal caries. It also has a buccal caries." However, there were cases where the tooth number was not stated in the same sentence as the finding. For example, in the sentences "Number 6 has distal decay. Next, we move to number 7. That tooth has an MOD amalgam." ONYX identified the distal decay on number six and then identified the MOD amalgam on tooth seven. Because there was no finding in the sentence "Next, we move to number 7," ONYX ignored that sentence and charted the MOD on tooth six resulting in a false positive.

5.3 DISCUSSION

In this study, we successfully created a speech-to-chart prototype for naturally dictated hard tissue dental exams. We evaluated the components of the prototype and found many areas that can be improved. Speech recognition is currently not robust enough to handle naturally dictated hard tissue exams. However, our simple post-processing techniques, although ad-hoc, illustrate

the potential for speech recognition adaptability to the dental domain. The charts created by ONYX from manually transcribed exams were significantly more accurate than charts created from automatically transcribed exams even after post-processing corrections were applied. Nevertheless, without improvements to speech recognition, our system cannot chart with the accuracy that dentists require. Overall, we were able to piece together an out-of-the-box speech recognizer with our NLP application and create a simple graphical chart generator that takes dental dictations and charts them in leading dental software. The flaws in our prototype highlight the areas we intend to enhance to build a more accurate and usable speech-to-chart system.

5.3.1 Improving speech recognition for the dental domain

The improvement of the post-processed exams, however small, suggests that implementation of a domain dictionary and a well-trained language model of bigrams has great potential to improve speech recognition performance. Based on the promising results of our ad-hoc post-processing methods, we worked with M*Modal on a preliminary adaptation of their speech recognizer. M*Modal is a company that produces speech understanding applications and offers conversational speech and natural language understanding services to healthcare providers. Their speech understanding technology analyzes physicians' free-form dictation recordings and encodes clinical concepts and their modifiers and relationships into structured documents that can be imported into Electronic Health Record systems [25, 26]. As a baseline, we implemented their acoustic model and medicine language model that they use to transcribe medical dictations. We then adapted the speech recognizer in several ways, including developing a language model with dental exams, enhancing the medical language with dental exams, and enhancing the

dictionary with words from our training exams. We evaluated the adapted models on a blind set of six exams.

Accuracy for the baseline recognizer was very low at 41 percent, which points out the vast difference between dictated medical reports and dental exams. Accuracy increased to 66 percent by augmenting the language model with 28 dental exams and to 73 percent with 43 exams. A small dictionary extension also improved performance. The top-performing combination showed 76 percent accuracy over the six reports by combining the baseline language model with a language model from 43 exams and the dictionary extension. With this combination, accuracy ranged by report from 73 to 86 percent. This is a marked improvement over the 67 percent accuracy of Dragon's transcriptions. Results from this preliminary work suggest that simple adaptation techniques are quite successful, that more training data is helpful, but that there is still room for improvement.

5.3.2 Improving ONYX for dental charting

Given ONYX's small training set of 13 exams (including two dentists and a hygienist), performance was quite high. Since most of the false negatives were due to gaps in ONYX's training (48 percent), more training promises even better performance. The training process for ONYX is typically time-consuming and labor-intensive. Humans must manually annotate training exams extracting concepts and identifying relationships [41]. We have developed a tool to aid human annotators in the training process. The tool is integrated with ONYX, so that new annotations immediately become part of ONYX's knowledge base, and ONYX can aid in the annotation of new sentences. In this tool, templates and relations are automatically created for a new sentence based on previous training.

Many of ONYX's false positives are because it currently does not have a sophisticated discourse processor and therefore cannot distinguish between a finding and a treatment plan. False positives also occur when ONYX makes wrong assumptions. For example, if ONYX identifies a finding and a surface but does not see a tooth number, it uses the pervious tooth number assuming the dentist is continuing to dictate findings on that tooth. A more sophisticated discourse processor that requires trigger words like "also" in the case of "number 9 has distal caries, also a buccal amalgam" will increase ONYX's accuracy.

Many of ONYX's errors can be fixed with more training and enhancements to the discourse processor. ONYX was developed with a number of innovative ideas including: a symbolic language extended to include probabilistic and procedural elements; an integration of syntax and semantics that includes a semantically weighted, probabilistic context-free grammar; and an interpretation based both on a semantic network and on semantic information attached to the syntactic grammar [41]. Considering ONYX's early stage of development it performed reasonably well in this evaluation but must be extended to address challenges in extracting findings from spoken dental exams [41].

5.3.3 Improving the prototype

Our speech-to-chart prototype is the first of its kind in dentistry. It allows dentists to dictate an exam naturally as if they were dictating to an assistant. From a technical standpoint, this prototype could easily be turned into a working product for dentists to use for dictating and charting dental exams. From a usability standpoint, however, several improvements and advances would be required to provide a speech-driven charting system that dentists would us in place of charting on paper or dictating to an assistant.

To be beneficial to dentists, our system needs to chart more hard tissue findings and possibly periodontal findings and treatment plans. As we have proven the feasibility of the prototype, we can begin to enhance ONYX's semantic model to include more concepts and provide training cases to ONYX with exams that include periodontal findings and treatment plans. Next, to integrate into the current workflow of dentistry, our system needs to chart in real-time. To do this we plan to couple the speech recognizer with ONYX. The two systems can provide feedback to each other that will assist in the recognizer selecting the correct word and ONYX slotting it in the correct node of the model.

As discussed previously, speech recognition needs to be improved for the dental domain. Not only does the recognizer have to be adapted for the dental domain, but we need to consider the clinical environment where exams will be dictated. Dental dictations are very different from radiology dictations and therefore we cannot expect the high level of accuracy found in radiology transcripts. Dental dictations occur in noisy operatories, the clinician is wearing a mask, and the patient is present. The dentist is performing an actual procedure or exam during the dictation and is therefore preoccupied during the dictation, making some parts of the dictation choppy and fragmented. Consequently, we have explored the idea of imposing some constraints on what the dentist can say. A loosely structured input could increase accuracy of the speech recognizer and of the NLP system. The constrained version of our system could require the dentist to speak the tooth number followed by any observations or plans for that tooth before moving onto the next tooth number. We believe this constraint could improve performance.

Finally, for our system to be adopted it needs to work with any of the leading dental software systems. To do this, we must update our graphical chart generator system by adding routines to chart findings for leading dental software. Another solution would be to work with

one of the leading software companies and create a clinical system designed specifically for speech data entry. As stated in our background (Section 2.4), researchers [28, 29] have shown that speech applications should not be built "on top of" graphical user interfaces, but instead should be designed from scratch. Because our system uses natural language for dictations, it is inherently more usable than current dental speech interfaces. For example, a dentist does not have to say "conditions," "move down 8," (for moving to the caries item on a list of conditions) "move down four", "ok" to select "caries" from the conditions list; he can simply say "caries." However, when incorporating our system with current charting software, we still need to be aware of usability issues related to the speech interface. One feature specific to the design of speech systems is how the user confirms speech input. Our system could supply the dentist with real-time text or audio feedback. The dentist could read on the screen in large text what was just recognized, or the system could speak a summary of what was just recognized so he does not need to remove his attention from the patient to check the screen. For example, if the dentist says "there is an MOL amalgam on tooth number one", the system can speak "tooth 1 MOL amalgam." Another feature specific to speech interfaces is training the system to understand the speaker's voice. If the system routinely misrecognizes words, the system should allow the dentist to easily train specific words, possibly in real time. Features like these will ultimately increase accuracy and satisfaction with the system and are necessary considerations when adapting the system for use with current dental practice management systems.

Assuming these improvements can be successfully implemented and the system can perform at least the 80 percent accuracy it performs at for manually exams, we believe that dentists could use our system in place of charting on paper or dictating to an assistant.

5.3.4 Limitations

This study has limitations. ONYX was trained with two dentists and one hygienist, but the datasets used in the study included exams dictated by dental students who may have different dictation styles. Next, although our test set size was adequate for showing statistically significant differences between a speech transcript and a manual transcript, the test set was too small to use traditional post-processing error-correction techniques. For example, due to our limited number of exams and hence words, we could not successfully implement a word co-occurrence algorithm [101]. Even with these limitations we are excited about the successful development of our speech-to-chart prototype and its ability to chart naturally dictated hard tissue exams.

The next section of this dissertation compares the results from the two objectives. These comparisons are followed by an overall discussion, limitations section, future work section and final conclusions.

6.0 JUST FOR FUN

An ideal conclusion to this work would be a comparison of the prototype with existing systems. Because we used different datasets and different subjects, and because our analysis of dental charting systems involved dental students reading scripts rather than charting live patients, any conclusions drawn from a comparison would be tentative, at best. But we were curious about how the speech-to-chart prototype compared to existing systems, and we believe the comparison provided some insight into the potential utility of a speech-to-chart system. So, just for fun we investigated the following research questions:

RESEARCH QUESTION 1: Can the speech-to-chart prototype chart findings in less time than existing dental practice management systems?

RESEARCH QUESTION 2: Can the speech-to-chart prototype chart findings with fewer errors than existing dental practice management systems?

6.1.1 Research Question 1: Can the speech-to-chart prototype chart findings in less time than existing dental practice management systems?

A direct comparison of the time to chart findings in the prototype with the time necessary to chart findings in existing applications is difficult to make for many reasons. First, students using the PMSs were reading pre-written scripts which allowed them to speak faster than students who

were dictating findings while examining a real patient. Next, in the analysis of existing charting systems, we were able to watch the video of the task and isolate the time it took for a student to chart an individual finding. The prototype's input was an entire dictation, often including descriptions of the patient and of findings outside the scope of hard tissue findings (e.g., PSR scores). Therefore, we could not easily isolate each finding in the natural dictations—as such, the calculations for time per finding were simple means. Finally, analysis of existing charting systems involved students charting the exact same nine hard tissue findings in each exam, whereas exams processed by the prototype were not standardized and contained an average of 28 findings per exam.

To calculate the average time per finding for the existing systems, we manually watched each exam video and timed how long it took to chart each of the nine hard tissue findings. We then averaged time per finding over all subjects to calculate average time per finding. Average time for existing systems is probably an underestimate of actual time to chart in a real exam, because subjects were provided with scripts for charting and did not have to remember commands themselves or make corrections based on mistakes they made in deciding which command to use. To calculate average time per finding for the prototype, we timed each exam dictation and subtracted any time spent speaking about things outside of hard tissue charting (e.g. patient identifiers or periodontal findings). Total time was calculated as the sum of dictation time and the time needed to process the exam with ONYX and chart it in Dentrix. The total time does not include the time taken to transcribe the exams, because in a real implementation, the transcription would occur automatically in real-time. We divided the total time required for each exam by the number of findings in that exam to get the mean time per finding for that exam and averaged those results over all exams to determine the average time to chart a single hard tissue

finding with the prototype. Like the measurements of average time for existing charting systems, average time for the speech-to-chart prototype is a low estimate compared to what would occur in a real setting, because the calculation did not account for corrections the dentist will ultimately have to make in response to charting errors resulting from mistakes in speech recognition output and in ONYX's interpretations.

When we compared Dentrix and EagleSoft in the performance evaluations (Section 4.2), the average time to chart a single hard tissue finding was 17.9 seconds. For the speech-to-chart prototype, the average time to chart a single hard tissue finding with the manually-transcribed exams was 7.3 seconds. Having described a number of caveats in this comparison, we still observe that the speech-to-chart prototype can chart findings in less time than existing dental practice management systems.

6.1.2 Research Question 2: Can the speech-to-chart prototype chart findings with fewer errors than existing dental practice management systems?

For similar reasons as stated in Section 6.1.1, is difficult to make direct comparisons between the errors made by existing systems and those made by the prototype. In the existing systems, students were charting the exact same nine findings for every exam. Further, they were not speaking naturally, so misrecognitions of words/phrases like "move down 9" will never appear in the prototype. However, to answer the research question, we calculated accuracy for Dentrix and EagleSoft—the only two systems that allowed hard tissue charting. We counted the number speech commands that did not result in an error (repeat, misrecognition, and insertion) during hard tissue charting for each exam and divided that by the number of hard tissue speech commands in each script. We calculated accuracy for the prototype, by totaling the true positives

with the true negatives and dividing that number by the total number of findings (including the correct true negatives). As with measures of time per finding, these measures of accuracy underestimate actual accuracy numbers.

The average accuracy for Dentrix and EagleSoft when reading from the script was 88 percent. For the speech-to-chart prototype, the average accuracy with Dragon was 48 percent, with Dragon and post-processing 54 percent, and with manual exams 80 percent. These results suggest that our speech-to-chart prototype charts exams with more errors than existing dental practice management systems. This finding supports the need to improve the prototype's ability to accurately chart findings, because the number of errors in a live setting using speech recognition will ultimately be much higher, and accuracy less than 80 percent may not be acceptable by dentists as an alternative to dictating to an assistant or manually charting on paper, in spite of the ease of dictation.

7.0 OVERALL DISCUSSION

7.1.1 Evaluation of existing speech-drive charting systems

For objective 1, we evaluated the efficiency, effectiveness, and user satisfaction of the speech interfaces of four dental practice management systems. Our results showed that practice management systems are attempting to accommodate speech recognition as a means of interaction. However, the existing systems have many limitations with speech functionality which may hinder their use. Through our findings, we can conclude that the current state of speech recognition for charting in dental software systems is insufficient for use during initial dental exams. Clearly, charting can be accomplished via speech in these systems—however three of the four systems required the use of the mouse and keyboard during charting and two of the systems did not support charting hard tissue findings via speech. Moreover, participants who used the systems articulated that the structured input necessary to chart in all four systems would be difficult to learn and uncomfortable to use.

The analysis of existing systems shows a need for a natural language interface that will allow clinicians to speak naturally as a means of entering data in a computer-based patient record without using the keyboard and mouse and without relying on an auxiliary. The absence of a flexible, robust, and accurate natural language interface is a significant barrier to the direct use of computer-based patient records by dental clinicians.

7.1.2 Speech-to-chart prototype

To address the need identified by objective 1, for objective 2 we developed and evaluated a speech-to-chart prototype for charting naturally spoken dental exams. Our system performed at 80 percent accuracy with manually transcribed exams and 54 percent accuracy with processed exams. Using manually-transcribed exams we were able to show that we could create an alternate end-to-end speech and natural language processing digital charting system that performed with accuracy similar to that of existing dental practice management systems. From a dentist's perspective, a clinician would have to correct approximately 20 percent of the findings in each exam. However, many of these corrections would not be correcting the entire finding, but only the surfaces.

Our speech-to-chart prototype is the first of its kind in dentistry. It allows dentists to dictate an exam naturally as if they were dictating to an assistant. The final chart taking approximately three-and-a-half minutes to complete under the ideal circumstance of perfect speech recognition appears to be less than the time to chart in the leading dental software systems under the ideal circumstance of having each command scripted for the user. The graphical chart generator that we created could easily be altered to work with any Microsoft Windows-based software program. From a technical standpoint, this prototype could easily be turned into a working product for dentists to use for dictating and charting dental exams. From a usability standpoint, however, several improvements and advances would be required to provide a speech-driven charting system that dentists would us in place of charting on paper or dictating to an assistant.

To be beneficial to dentists, our system needs to chart the majority of hard tissue findings, periodontal findings, and possibly treatment plans. As we have proven the feasibility of

the prototype, we can now enhance ONYX's semantic model to include more concepts as well as providing training cases to ONYX which include periodontal findings and treatment plans. Ideally, if our system is adopted by multiple dentists, training would occur as the system was used.

Next, our system can be designed to share information as a part of the emerging National Health Information Infrastructure (NHII) [102]. The NHII is an initiative set by the U.S. Department of Health and Human Services to connect health information via interoperable systems across the U.S. [103]. The NHII's goal is to improve the effectiveness, efficiency and quality of health care information and improve clinical decision-making by making health information easily accessible [103]. We need to design the system to support data sharing so our software can participate in data repositories when dental software systems are connected across the U.S. Theoretically, all dentists using our system will be able to share their clinical data, allowing our system to take unique advantage of this larger, distributed dataset to enhance training and increase accuracy. When a dentist uses our system and trains it on new concepts or new terms, this data will be instantly available to the systems being used by all other dentists participating in data sharing. For example, if the baseline system that all dentists use has not been trained on the phrase "not there"—as in "tooth one is not there"—and one dentist trains the system that this new phrase maps to the concept "missing tooth," all other systems would then be able to map this phrase to the correct concept. Incorporating the ability to share data into our system would exponentially expand the number of training cases that can be used to enhance system performance.

Our system can also use data repositories thorough the NHII to enhance local customization. As described in Section 2.8, ONYX uses probabilistic models to learn

relationships between words and concepts. Thus, each dentist using our system would have a unique probability distribution based on the types of patient cases he sees. In this way, the system would be tailored to each dentist's individual practice. For example, a pediatric dentist may see a significantly lower number of caries cases and higher number of sealants than a general dentist. As such, for the pediatric dentist, ONYX may be more likely to slot ambiguous terms into sealant slots rather than caries slots. Hence, data repositories through the NHII offer a combination of local customization and distributed training data, which would enhance performance of individual versions of ONYX.

Next, for usability, our system needs to integrate into the current workflow of dentistry—that is, it should chart in real-time. To do this we plan to couple the speech recognizer with ONYX. The two systems can provide feedback to each other that will assist in the recognizer selecting the correct word and ONYX slotting it in the correct node of the model. Finally, our graphical chart generator should be updated to work with the most current version of Dentrix or we should work with Dentrix to incorporate our system into their software to allow for tailored speech input functionality.

Assuming these improvements can be successfully implemented and the system can perform at least the 80 percent accuracy it performs at for manually exams, we believe that dentists could use our system in place of charting on paper or dictating to an assistant.

7.1.3 Limitations

This work has many limitations. In our feature analysis and performance evaluations we only compared four of the leading dental charting systems. We know of at least one more system (Mogo Dental Software, Westmont, IL) that advertises a speech interface for dental charting.

However, acquiring all possible dental systems for comparison was beyond the ability of our research staff. Further, this study took place over the course of multiple years. Therefore, each of the four systems has released newer versions of their software and thus may have made improvements to their speech interface design. We used a convenience sample of dentals students from the University of Pittsburgh who all spoke American-English as their native language. A random sample of students from dental schools across the country may have provided more generalizable results.

Our speech-to-chart prototype was created to test the feasibility of such a system. Therefore, all aspects of the system were underdeveloped. First, the prototype does not work in real-time. Dictations are transcribed by Dragon Naturally Speaking and those transcriptions are then processed and charted. We have future plans to incorporate the speech recognizer with the NLP application. As for the NLP application, ONYX itself is a new system that was built for this project. ONYX has only been trained on 12 hard tissue exams from two dentists and one hygienist. Also, ONYX's semantic model was only designed for 13 of the most common hard tissue findings—although that number is being extended. ONYX is currently not publically available and can only be accessed within the University of Pittsburgh's Department of Biomedical Informatics—however after further improvements, we plan to make ONYX available with open source licensing. Finally, our datasets for the development and evaluation of our prototype appear small with a total of 25 exams from six dental students. Even though our sample size calculations in Appendix A show that the number of findings from these exams are more than adequate for the statistical analyses we performed, a greater number of exams from a broader range of dental clinicians would make our results more generalizable.

7.1.4 Future Work

We have many plans to extend our work and improve our speech-to-chart prototype. First, we will continue to gather exams from dentists and dental students to enhance ONYX. More training data will increase ONYX's accuracy and allow us to expand the semantic model to include all hard tissue findings, periodontal findings, and treatment plans. We plan to integrate the speech recognizer with the NLP system so that charting can occur in real-time. The speech recognizer should be able to pass sentences or individual findings to the NLP application which can be charted as they are dictated. To accomplish this, we will work with M*Modal [25, 26] in integrating more dental exams into the language models to improve speech recognition accuracy. Finally, we will continue developing ONYX and will make it publically available.

8.0 CONCLUSIONS

In this dissertation we were able to show that existing speech interfaces for dental software are less than ideal. We were able to point out the reasons they were inadequate—the main flaw being that they require the dentist to use very structured commands to interact with the system. We showed that Dentrix was the system with the most robust speech interface capabilities. However, none of the leading systems—including Dentrix—allowed the dentist to chart using natural language akin to their current way of dictating exams. Therefore, we aimed to create a system that can facilitate natural speech input. In this dissertation, we successfully created a speech-to-chart prototype which can chart naturally-spoken exams. We evaluated performance of an existing speech recognition system on charting dental exams and showed that much work is needed to automatically generate accurate transcriptions. With accurate transcriptions, we showed that ONYX could chart findings described in the transcriptions with fairly good accuracy, especially considering its early stage of development. This dissertation work brings us closer to providing dentists with a natural-language interface to interact with the clinical computer at chairside.

Dentistry is currently in an exciting time, over the last several years, there has been a significant development of new technologies with the number of computer-based devices in the dental office skyrocketing [86]. Technology has become the center of many practices where especially in the administrative areas, computers are all but ubiquitous [86]. Technology is also

gaining a significant presence in clinical dentistry as well [86]. Chairside computing has adopted technology at a slower rate for many reasons including technology hindering the clinical workflow [104], cumbersome systems [8], and lack of integration [86]. Our prototype addresses all of these limitations. We anticipate our system and systems like it to enhance clinical care by providing dentists with technology designed according to their needs. Only then can dentists realize the benefits of improved documentation, increased efficiency, and chairside decision support for enhanced diagnoses, treatment planning, and overall patient care.

APPENDIX A

SAMPLE SIZE CALCULATIONS FOR SUMMATIVE EVALUATIONS OF SPEECH-TO-CHART PROTOTYPE

To make comparisons of accuracies between exams in the speech-to-chart summative evaluation we completed sample size and power calculations. The expected sample size is calculated using average exam findings from four preliminary exams. The four preliminary exams had an average of 31 findings per exam. Therefore, to estimate sample size of our 12 exams (Test Set) we multiplied 12 x 31 findings = 372 chartable conditions. In our study, we calculated significance of difference in accuracy between the three types of exams: manual transcriptions, Dragon transcriptions and Dragon plus post-processing transcriptions. To get a crude sense of the estimated number of findings and the statistical power needed for detecting differences, we consider a McNemar's test with the accuracies from the preliminary data:

$$n = \frac{\left(Z_{1-\frac{\alpha}{2}} + 2 Z_{1-\beta} \sqrt{P_A Q_A} \right)^2}{4\left(P_A - 0.5\right)^2 P_D} \qquad Power = \Phi\left[\frac{1}{2\sqrt{P_A Q_A}} \left(Z_{1-\frac{\alpha}{2}} + 2\left|P_A - 0.5\right|\sqrt{nP_D} \right) \right]$$

Where $\alpha = 0.005$, so that $Z_{1-\frac{\alpha}{2}} = 1.96$ and $\beta = 0.05$, so that $Z_{1-\beta} = 1.645$ (or a power of 95%). P_A and Q_A = the proportions of discordant and P_D = the proportion of I deleted this part, because I

don't understand how this was practically carried out – you didn't count false negatives and then subtract all but 32, did you? discordant over all findings from the preliminary exams (n=118).

The results of the McNemar's test can be found in Table 15. We are well aware that McNemar's test involves the assumption of independence of findings. We know that the numbers in Table 15 ignore any data dependencies from individual patients or the dental students who completed the exams and may make the estimated sample size and power considerably over-optimistic. For that reason the actual sample size (n=338) is much larger than the naïve sample size and power calculations would call for. In addition to measuring accuracy, we measured sensitivity, specificity, positive predictive value, and negative predictive value.

Table 15. Sample size and power calculations (n=372) at a confidence level of 0.95. Dragon transcripts (D), Dragon with post-processing routines (D+PP), and manually transcribed transcripts (MT).

	accuracy comparisons		
	MT & D	**MT & DPP**	**D & DPP**
estimated n necessary	20.3	9.2	20.2
Power	0.99	1.0	0.99

APPENDIX B

POST-PROCESSING ERROR CORRECTION CODE WRITTEN IN PYTHON

```
'''Created on Jun 11, 2009, post-processing algorithm, author:
Jeannie Irwin'''

import glob
import re
import string

def spelling(doc, changes):
    """Replaces common spelling mistakes and homophones. Returns
    the updated document and number of changes made."""
    spellDict={"carries":"caries","buckle":"buccal",
    "peter":"pieter"}
    z=0 # index of word location
    outcomes=[]
    for word in doc:
        if word in spellDict:
            doc[z]=spellDict.get(word)
            print "changing", word, "to", doc[z]
            changes+=1
        z=z+1
        newdoc=""
        for y in doc:
            newdoc=newdoc+y+" "
        newdoc=newdoc+"\n"
        newdoc=newdoc.lstrip()

    outcomes.append(newdoc)
    outcomes.append(changes)
    return(outcomes)

def nForM(file, changes):
```

```
"""Replaces commonly mis-transcribed words. Returns the
updated document and number of changes made."""
outcomes=[]
nForMdict={"and ongoing":"amalgam","thistle":"distal","to
cave":"decayed", "civilian":"severly", "means
you'll":"mesial", "ammo":"mo", "ammount on":"amalgam",
"posting":"post and", "can now":"canal", "serve a":"survey",
"clues will":"occlusal", "clues old":"occlusal", "gop":"dob",
"buck whole":"buccal", "buckled":"buccal", "malvo":"amalgam",
"a clue soul": "occlusal", "amount rome": "amalgam", "next
layer": "maxillary", "amount of":"amalgam", "3
current":"recurrent","bk":"decay","pistol":"distal", "can
pause it":"composite", "aid":"8", "recount":"root canal",
"as":" has", "dk":"decay", "when he":"20", "clues
oh":"occlusal","musical":"mesial","book will":"buccal",
"kerry is":"caries", "kerry's":"caries","outcome":"amalgam",
"dumb":"number", "k":"decay", "ground":"crown",
"label":"lingual", "lethal":"lingual","ripped now":"root
canaled", "compulsive":"composite", "into
less":"endentulous", "politics":"pontics", "tubercle":"2
buccal", "sozzled":"incisal", "official":"facial",
"nonofficial":"facial", "fish":"facial", "amount
on":"amalgam", "a malvo":"amalgam", "his still":"distal",
"paid horseman":"porcelain", "spine":"fine", "amount will":
"amalgam", "president":"present", "mine":"fine",
"hosting":"post and", "oh":"o", "disco":"distal",
"visual":"facial", "posters":"posteriors", "max
o'leary":"maxillary", "max larry":"maxillary", "demand a
bull": "mandible", "maysville":"mesial", "a chill":"facial",
"need still":"mesial","detained":"decay", "rating
graph":"radiograph","this will":"distal","close will":
"occlusal", "me feel":"mesial", "fiscal":"distal", "unmount
on": "amalgam", "bissell":"distal", "and size of":"incisal",
"media":"mesial"}
doc=file.split()
newdoc=""
newSen=""
for word in doc: # this loop replaces single word errors
    ItemKeys=nForMdict.keys()
    if word in ItemKeys:
        newSen=newSen+nForMdict.get(word)+" "
        print "replacing", word, "with", nForMdict.get(word)
        changes+=1
    else:
        newSen=newSen+word+" "
newdoc=newdoc+newSen+". "
```

```
    again=newdoc.split(".")  # this loop replaces multiple word errors
    newdoc2=""
    ItemKeys2=nForMdict.keys()
    keyList=[]
    for item in ItemKeys:
        newItem=item.split()
        if len(newItem)>1:  #get errors from dict with two or more words
            keyList.append(string.join(newItem, ' ' ))
    for sen in again:
        newSen2=""
        for y in keyList:
            if y in sen:
                newSen2=sen.replace(y, nForMdict.get(y)+" ")
                sen=newSen2
                print "replacing", y, "with", nForMdict.get(y)
                changes+=1
            else:
                newSen2=sen
        newdoc2=newdoc2+newSen2+". "
    newdoc2=newdoc2.lstrip()
    outcomes.append(newdoc2)
    outcomes.append(changes)
    return(outcomes)

def ngram(file,changes):
    """Replaces context-based spelling and homophone errors.
    Returns the updated document and number of changes made."""
    newdoc=""
    outcomes=[]
    doc=file.split()
    numList=["1","2","3","4","5","6","7","8","9","10","11","12",
    "13", "14", "15", "16", "17", "18", "19", "20", "21", "22",
    "23", "24", "25", "26", "27", "28", "29", "30", "31", "32"]
    z=0
    while z<len(doc):
        if doc[z] == "to":
        #print  doc[z-1], doc[z], doc[z+1]
            if doc[z+1] in numList and doc[z-1] not in numList:
               print "replacing", doc[z], "with tooth,
               sentence:", doc[z-1], doc[z], doc[z+1]
               doc[z]="tooth"
               changes+=1
            elif doc[z+1]=="number":
               print "replacing", doc[z], "with tooth,
               sentence:", doc[z-1], doc[z], doc[z+1], doc[z+2]
               doc[z]="tooth"
               changes+=1
```

```
        elif doc[z+1]==".":
           print "replacing", doc[z], "with 2, sentence:",
           doc[z-1], doc[z], doc[z+1]
           doc[z]="2"
           changes+=1
        elif doc[z-1]=="tooth":
           print "replacing", doc[z], "with 2, sentence:",
           doc[z-1], doc[z], doc[z+1]
           doc[z]="2"
           changes+=1
        z+=1
    else: z+=1

z=0
while z<len(doc):
    if doc[z] == "for":
        #print doc[z-2], doc[z-1], doc[z], doc[z+1],
        doc[z+2]
        if doc[z-1]=="tooth":
           #print "replacing", doc[z], "with 4, sentence:",
           doc[z-2], doc[z-1], doc[z], doc[z+1], doc[z+2]
           doc[z]="4"
           changes+=1
        elif doc[z-1]=="number":
           #print "replacing", doc[z], "with 4, sentence:",
           doc[z-2], doc[z-1], doc[z], doc[z+1], doc[z+2]
           doc[z]="4"
           changes+=1
        z+=1
    else: z+=1


z=0
while z<len(doc):
    if doc[z] == "and":
      #print doc[z-2], doc[z-1], doc[z], doc[z+1], doc[z+2]
      if doc[z-1]=="tooth":
           #print "replacing", doc[z], "with 10, sentence:",
           doc[z-2], doc[z-1], doc[z], doc[z+1], doc[z+2]
           doc[z]="10"
           changes+=1
      elif doc[z-1]=="number":
           #print "replacing", doc[z], "with 10, sentence:",
           doc[z-2], doc[z-1], doc[z], doc[z+1], doc[z+2]
           doc[z]="10"
           changes+=1
       z+=1
```

```
        else: z+=1

    for item in doc:
        newdoc=newdoc+item+" "
    outcomes.append(newdoc)
    outcomes.append(changes)
    return(outcomes)

def getFiles(oDir):
    """Opens exams transcribed by Dragon. Pass the function the
    original directory containing the files. Returns a list of
    file names."""
    folder=oDir
    loc=folder+"drag_*.txt"
    print "opening files from here: ", loc
    txtlist=glob.glob(loc)
    return txtlist

def writeFiles(item, FinalString):
    """Writes new exams. Pass the function the original file name
    and the new file."""
    newdir=item.split("\\")
    newdst=newdir[0]+"\\new\\"+newdir[1]
    print "saving edited file here: ", newdst
    f=open(newdst, 'w')
    FinalString=str(FinalString)
    f.write(FinalString)
    f.close()

def main():
    textlist=getFiles("test/")
    changesSpell=0 #counter for number of changes made by spelling function
    changesNfM=0 #counter for number of changes made by nforM function
    changesNgram=0 #counter for number of changes made by n-gram function
    for item in textlist:
        infile= open(item, 'r')
        doc=infile.read()
        doc=doc.lower()
        doc=doc.split()
        infile.close()
        print item

        x=spelling(doc, changesSpell)
        changesSpell=x[1]
        spellString=x[0]

        z=nForM(spellString, changesNfM)
```

```
        changesNfM=z[1]
        nFmString=z[0]

        q=ngram(nFmString, changesNgram)
        changesNgram=q[1]
        FinalString=q[0]

        writeFiles(item, FinalString)
        print "----------------------------"

    print "total spelling changes made: ", changesSpell
    print "total nForM changes made: ", changesNfM
    print "total ngram changes made: ", changesNgram

if __name__== "__main__":
     main()
```

BIBLIOGRAPHY

[1] Borowitz S. Computer-based speech recognition as an alternative to medical transcription. Journal of American Medical Informatics Association. 2001;8(1):101-2.

[2] Devine E, Gaehde S, Curtis A. Comparative evaluation of three continuous speech recognition software packages in the generation of medical reports. Journal of American Medical Informatics Association. 2000;7(5):462-8.

[3] Grasso M, editor. The long-term adoption of speech recognition in medical applications. Computer-Based Medical Systems, 2003 Proceedings 16th IEEE Symposium; 2003.

[4] White K. Speech recognition implementation in radiology. Pediatric Radiology. 2005;35(9):841-6.

[5] Zafar A, Overhage J, McDonald C. Continuous speech recognition for clinicians. Journal of American Medical Informatics Association. 1999;6(3):195-204.

[6] Zick R, Olsen, J. Voice recognition software versus a traditional transcription service for physician charting in the ED. American Journal of Emergency Medicine. 2001;19:295-8.

[7] Schleyer T, Thyvalikakath T, Spallek H, Torres-Urquidy M, Hernandez P, Yuhaniak J. Clinical computing in general dentistry. Journal of the American Medical Informatics Association. 2006;13(3):344-52.

[8] Yuhaniak Irwin J, Fernando S, Schleyer T, Spallek H. Speech recognition in dental software systems: features and functionality. Stud Health Technol Inform. 2007;129(Pt 2):1127-31.

[9] Doolan DF, Bates DW. Computerized physician order entry systems in hospitals: mandates and incentives. Health Aff (Millwood). 2002 Jul-Aug;21(4):180-8.

[10] Drevenstedt G, McDonald J, Drevenstedt L. The role of voice-activated technology in today's dental practice. Journal of American Dental Association. 2005;136(2):157-61.

[11] American Dental Association Survey. 2006 Technology Survey. Chicago, IL2007.

[12] Schleyer T, Spallek H, Hernandez P. A qualitative investigation of the content of dental paper-based and computer-based patient record formats. Journal of the American Medical Informatics Association. 2007 Jul-Aug;14(4):515-26.

[13] Peacocke R, Graf D. An Introduction to Speech and Speaker Recognition. Computer. 1990;23(8):26-33.

[14] Young S. A review of large-vocabulary continuous-speech recognition. IEEE Signal Processing Magazine. 1996:45-57.

[15] Jurafsky D, Martin J. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition. Upper Saddle River, NJ: Prentice Hall; 2000.

[16] Voll K, Atkins S, Forster B. Improving the Utility of Speech Recognition Through Error Detection. Journal of Digital Imaging. 2008;21(4):371-7.

[17] Clayton PD, Naus SP, Bowes WA, Madsen TS, Wilcox AB, Orsmond G, et al. Physician use of electronic medical records: issues and successes with direct data entry and physician productivity. AMIA Annu Symp Proc. 2005:141-5.

[18] Schleyer TK, Thyvalikakath TP, Malatack P, Marotta M, Shah TA, Phanichphant P, et al. The feasibility of a three-dimensional charting interface for general dentistry. Journal of American Dental Association. 2007 Aug;138(8):1072-80.

[19] Devine E, Gaehde S, Curtis A. Comparative evaluation of three continuous speech recognition software packages in the generation of medical reports. Journal of American Medical Informatics Association. 2000;7(5):462-8.

[20] Ilgner J, Duwel P, Westhofen M. Free-text data entry by speech recognition software and its impact on clinical routine. Ear Nose Throat Journal. 2006 Aug;85(8):523-7.

[21] Issenman RM, Jaffer IH. Use of voice recognition software in an outpatient pediatric specialty practice. Pediatrics. 2004;114(3):290-3.

[22] Pezzullo JA, Tung GA, Rogg JM, Davis LM, Brody JM, Mayo-Smith WW. Voice Recognition Dictation: Radiologist as Transcriptionist. Journal of Digital Imaging. 2007 Jun 7.

[23] Lacson R, Barzilay R. Automatic processing of spoken dialogue in the home hemodialysis domain. Proc AMIA Annual Fall Symp. 2005:420-4.

[24] Happe AP, B, Burgun A, Cuccia M, Le Beux P. Automatic concept extraction from spoken medical reports. International Journal of Medical Informatics. 2003;70:255-63.

[25] Fritsch J. Using Speech Understanding to Improve Transcription Margins. MultiModal White Paper; 2006.

[26] Fritsch J. Utilizing Structured Narrative to Advance EHR Development.: MultiModal White Paper; 2008.

[27] Raskin J. The humane interface: new directions for designing interactive systems. 1 ed: Addison-Wesley Professional; 2000.

[28] Yankelovich N, Levow G, Marx M, editors. Designing SpeechActs: Issues in Speech User Interfaces. Conference on Human Factors in Computing Systems; 1995; Chicago, IL.

[29] Harris R. Voice interaction design: Morgan Kaufmann; 2005.

[30] Cohen T, Blatter B, Patel V. Exploring dangerous neighborhoods: latent semantic analysis and computing beyond the bounds of the familiar. Proc AMIA Annual Fall Symp. 2005:151-5.

[31] Fiszman M, Chapman WW, Aronsky D, Evans RS, Haug PJ. Automatic detection of acute bacterial pneumonia from chest X-ray reports. Journal of American Medical Informatics Association. 2000 Nov-Dec;7(6):593-604.

[32] Sinha U, Ton A, Yaghmai A, Taira RK, Kangarloo H. Image content extraction: application to MR images of the brain. Radiographics. 2001 Mar-Apr;21(2):535-47.

[33] Xu R, Garten Y, Supekar KS, Das AK, Altman RB, Garber AM. Extracting subject demographic information from abstracts of randomized clinical trial reports. Stud Health Technol Inform. 2007;129(Pt 1):550-4.

[34] Morioka CA, Sinha U, Taira R, el-Saden S, Duckwiler G, Kangarloo H. Structured reporting in neuroradiology. Ann NY Acad Sci. 2002 Dec;980:259-66.

[35] Jacquemart P, Zweigenbaum P. Towards a medical question-answering system: a feasibility study. Stud Health Technol Inform. 2003;95:463-8.

[36] Rosemblat G, Gemoets D, Browne AC, Tse T. Machine translation-supported cross-language information retrieval for a consumer health resource. AMIA Annu Symp Proc. 2003:564-8.

[37] Zeng QT, Goryachev S, Weiss S, Sordo M, Murphy SN, Lazarus R. Extracting principal diagnosis, co-morbidity and smoking status for asthma research: evaluation of a natural language processing system. BMC Medical Informatics and Decision Making. 2006;6:30.

[38] Coden A, Savova G, Sominsky I, Tanenblatt M, Masanz J, Schuler K, et al. Automatically extracting cancer disease characteristics from pathology reports into a Disease Knowledge Representation Model. Journal of Biomedical Informatics. 2008;Dec 27.

[39] Mutalik PG, Deshpande A, Nadkarni PM. Use of general-purpose negation detection to augment concept indexing of medical documents: a quantitative study using the UMLS. Journal of the American Medical Informatics Association. 2001 Nov-Dec;8(6):598-609.

[40] Chapman WW, Bridewell W, Hanbury P, Cooper GF, Buchanan BG. A simple algorithm for identifying negated findings and diseases in discharge summaries. Journal of Biomedical Informatics. 2001 Oct;34(5):301-10.

[41] Christensen L, Harkema H, Haug P, Irwin J, Chapman W. ONYX: A System for the Semantic Analysis of Clinical Text. BioNLP, A workshop of NAACL-HLT; Boulder, Colorado; 2009.

[42] Friedman C, Hripcsak G. Natural language processing and its future in medicine. Acad Med. 1999 Aug;74(8):890-5.

[43] Spyns P. Natural language processing in medicine: an overview. Methods Inf Med. 1996 Dec;35(4-5):285-301.

[44] Friedman C. A broad-coverage natural language processing system. Proc AMIA Symp. 2000:270-4.

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

[46] Friedman C, Hripcsak G, Shablinsky I. An evaluation of natural language processing methodologies. Proc AMIA Symp. 1998:855-9.

[47] Friedman C, Hripcsak G, Shagina L, Liu H. Representing information in patient reports using natural language processing and the extensible markup language. Journal of the American Medical Informatics Association. 1999 Jan-Feb;6(1):76-87.

[48] Hripcsak G, Austin JH, Alderson PO, Friedman C. Use of natural language processing to translate clinical information from a database of 889,921 chest radiographic reports. Radiology. 2002 Jul;224(1):157-63.

[49] Hripcsak G, Friedman C, Alderson PO, DuMouchel W, Johnson SB, Clayton PD. Unlocking clinical data from narrative reports: a study of natural language processing. Annals of Internal Medicine. 1995 May 1;122(9):681-8.

[50] Imai T, Aramaki E, Kajino M, Miyo K, Onogi Y, Ohe K. Finding malignant findings from radiological reports using medical attributes and syntactic information. Stud Health Technol Inform. 2007;129(Pt 1):540-4.

[51] Taira RK, Soderland SG. A statistical natural language processor for medical reports. Proc AMIA Symp. 1999:970-4.

[52] Taira RK, Soderland SG, Jakobovits RM. Automatic structuring of radiology free-text reports. Radiographics. 2001 Jan-Feb;21(1):237-45.

[53] Aronsky D, Fiszman M, Chapman WW, Haug PJ. Combining decision support methodologies to diagnose pneumonia. Proc AMIA Symp. 2001:12-6.

[54] Mitchell KJ, Becich MJ, Berman JJ, Chapman WW, Gilbertson J, Gupta D, et al. Implementation and evaluation of a negation tagger in a pipeline-based system for information extraction from pathology reports. Medinfo. 2004;2004:663-7.

[55] Pakhomov S, Buntrock J, Duffy P. High throughput modularized NLP system for clinical text. Proc ACL interactive poster and demonstration sessions. 2005:25-8.

[56] Chapman WW, Christensen LM, Wagner MM, Haug PJ, Ivanov O, Dowling JN, et al. Classifying free-text triage chief complaints into syndromic categories with natural language processing. Artificial Intelligence in Medicine. 2005 Jan;33(1):31-40.

[57] Hripcsak G, Knirsch CA, Jain NL, Stazesky RC, Jr., Pablos-Mendez A, Fulmer T. A health information network for managing innercity tuberculosis: bridging clinical care, public health, and home care. Computers and Biomedical Research. 1999 Feb;32(1):67-76.

[58] Ivanov O, Gesteland P, Hogan W, Mundorff MB, Wagner MM. Detection of Pediatric Respiratory and Gastrointestinal Outbreaks from Free-Text Chief Complaints. Proc AMIA Annu Fall Symp. 2003:318-22.

[59] Ivanov O, Wagner MM, Chapman WW, Olszewski RT. Accuracy of three classifiers of acute gastrointestinal syndrome for syndromic surveillance. Proc AMIA Symp. 2002:345-9.

[60] Jain NL, Knirsch CA, Friedman C, Hripcsak G. Identification of suspected tuberculosis patients based on natural language processing of chest radiograph reports. Proc AMIA Annu Fall Symp. 1996:542-6.

[61] Knirsch CA, Jain NL, Pablos-Mendez A, Friedman C, Hripcsak G. Respiratory isolation of tuberculosis patients using clinical guidelines and an automated clinical decision support system. Infection Control and Hospital Epidemiology. 1998 Feb;19(2):94-100.

[62] Fiszman M, Haug PJ, Frederick PR. Automatic extraction of PIOPED interpretations from ventilation/perfusion lung scan reports. Proc AMIA Symp. 1998:860-4.

[63] Chapman WW, Fiszman M, Frederick PR, Chapman BE, Haug PJ. Quantifying the characteristics of unambiguous chest radiography reports in the context of pneumonia. Academic Radiology. 2001 Jan;8(1):57-66.

[64] Sinha U, Taira R, Kangarloo H. Structure localization in brain images: application to relevant image selection. Proc AMIA Symp. 2001:622-6.

[65] Sinha U, Dai B, Johnson DB, Taira R, Dionisio J, Tashima G, et al. Interactive software for generation and visualization of structured findings in radiology reports. American Journal of Roentgenology. 2000 Sep;175(3):609-12.

[66] Wilcox AB, Narus SP, Bowes WA, 3rd. Using natural language processing to analyze physician modifications to data entry templates. Proc AMIA Symp. 2002:899-903.

[67] Lovis C, Chapko MK, Martin DP, Payne TH, Baud RH, Hoey PJ, et al. Evaluation of a command-line parser-based order entry pathway for the Department of Veterans Affairs electronic patient record. Journal of the American Medical Informatics Association. 2001 Sep-Oct;8(5):486-98.

[68] Fiszman M, Haug PJ. Using medical language processing to support real-time evaluation of pneumonia guidelines. Proc AMIA Symp. 2000:235-9.

[69] Turchin A, Pendergrass ML, Kohane IS. DITTO - a tool for identification of patient cohorts from the text of physician notes in the electronic medical record. Proc AMIA Annual Fall Symp. 2005:744-8.

[70] Pakhomov S, Weston SA, Jacobsen SJ, Chute CG, Meverden R, Roger VL. Electronic medical records for clinical research: application to the identification of heart failure. American Journal of Managed Care. 2007 Jun;13(6 Part 1):281-8.

[71] Wilke RA, Berg RL, Peissig P, Kitchner T, Sijercic B, McCarty CA, et al. Use of an electronic medical record for the identification of research subjects with diabetes mellitus. Clinical Medicine and Research. 2007;5(1):1-7.

[72] Olasov B, Sim I. RuleEd, a web-based semantic network interface for constructing and revising computable eligibility rules. AMIA Annu Symp Proc. 2006:1051.

[73] Patrick J, Zhang Y, Wang Y. Developing Feature Types for Classifying Clinical Notes. BioNLP. 2007.

[74] Uzuner O, Goldstein I, Luo Y, Kohane I. Identifying patient smoking status from medical discharge records. Journal of the American Medical Informatics Association. 2008 Jan-Feb;15(1):14-24.

[75] Pakhomov SS, Hemingway H, Weston SA, Jacobsen SJ, Rodeheffer R, Roger VL. Epidemiology of angina pectoris: role of natural language processing of the medical record. American Heart Journal. 2007 Apr;153(4):666-73.

[76] Denny JC, Peterson JF. Identifying QT prolongation from ECG impressions using natural language processing and negation detection. Stud Health Technol Inform. 2007;129(Pt 2):1283-8.

[77] Hahn U, Romacker M, Schulz S. Discourse structures in medical reports--watch out! The generation of referentially coherent and valid text knowledge bases in the MEDSYNDIKATE system. International Journal of Medical Informatics. 1999 Jan;53(1):1-28.

[78] Hahn U, Romacker M, Schulz S. MEDSYNDIKATE-a natural language system for the extraction of medical information from findings reports. International Journal of Medical Informatics. 2002 Dec;67(1-3):63-74.

[79] Rindflesch TC, Fiszman M. The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text. Journal of Biomedical Informatics. 2003 Dec;36(6):462-77.

[80] Irwin J, Harkema H, Christensen L, Schleyer T, Haug P, Chapman W. Methodology to Develop and Evaluate a Semantic Representation for NLP. AMIA Annu Symp Proc. 2009.

[81] Quillian M. Semantic memory. In: Minsky M, editor. Semantic Information Processing. Cambridge, MA: MIT Press; 1968.

[82] Rocha RA, Huff SM, Haug PJ, Evans DA, Bray BE. Evaluation of a semantic data model for chest radiology: application of a new methodology. Methods Inf Med. 1998 Nov;37(4-5):477-90.

[83] Roberts A, Gaizauskas R, Hepple M, Demetriou G, Guo Y, Setzer A, et al. The clef corpus: semantic annotation of clinical text. AMIA Annu Symp Proc. 2007. p. 625-9.

[84] Christensen L, Haug PJ, Fiszman M. MPLUS: a probabilistic medical language understanding system. Proc Workshop on Natural Language Processing in the Biomedical Domain. 2002:29-36.

[85] Koehler SB. SymText: A natural language understanding system for encoding free text medical data. Salt Lake City: University of Utah; 1998.

[86] Schleyer T. Why integration is key for dental office technology. The Journal of the American Dental Association. 2004;135(Suppl):4S-9S.

[87] Schleyer TK, Torres-Urquidy H, Straja S. Validation of an instrument to measure dental students' use of, knowledge about, and attitudes towards computers. Journal of Dental Education. 2001 Sep;65(9):883-91.

[88] Hone KS, Graham R. Towards a tool for the subjective assessment of speech system interfaces (SASSI). Natural Language Engineering. 2000;6(3-4):287-303.

[89] Malkovich JF, Afifi AA. On Tests for Multivariate Normality. Journal of the American Statistical Association. 1973;68(341):176-9.

[90] Goodacre J, Nakajima Y. The Perception of Fricative Peaks and Noise Bands. Journal of Physiological Anthropology and Applied Human Science. 2005;24(1):151-4.

[91] Muchmore M. Dragon NaturallySpeaking 10. PC Magazine. 2008.

[92] Audacity: Free Audio Editor and Recorder. 2009 [August 14, 2009]; Available from: http://audacity.sourceforge.net/.

[93] Evaluation Tools. Information Technology Laboratory, National Institutes of Standards and Technology 2007 [updated September 12, 2008; cited 2008 September 19]; Available from: http://www.nist.gov/speech/tools/.

[94] Cuendet S, Hakkani-Tur D, Tur G. Model Adaptation for Sentence Segmentation from Speech, Spoken Language Technology Workshop. IEEE. 2006; p.102-5.

[95] Furui S. Recent advances in speaker recognition. Pattern Recognition Letters. 1997;18(9):859-72.

[96] Huang XD, Lee KF, Hon HW, Hwang MY. Improved acoustic modeling with the SPHINX speech recognition system. IEEE International Conference on Acoustics, Speech, and Signal Processing; Toronto, Ontario, Canada. 1991. p. 345-8.

[97] Schuler W, Wu S, Schwartz L. A Framework for Fast Incremental Interpretation during Speech Decoding. Computational Linguistics. 2009:1-31.

[98] Wang Z, Schultz T, Waibel A. Comparison of acoustic model adaptation techniques on non-native speech. Acoustics, Speech, and Signal Processing, 2003 Proceedings (ICASSP '03) IEEE International Conference on:540-3.

[99] Voll K. A methodology of error detection: Improving speech recognition in radiology [Ph.D. thesis]: Simon Fraser University; 2006.

[100] Chin A. Dentrix Dental Practice Management Software. DentalCompare.com; [september 14, 2009]; Available from: http://www.dentalcompare.com/review.asp?rid=7.

[101] Sarma A, Palmer D, editors. Context-based speech recognition error detection and correction. HLT-NAACL; 2004.

[102] Acharya A, Mital DP, Schleyer TK. Electronic dental record information model. International Journal of Medical Engineering and Informatics. 2009;1(4):418 - 34

[103] Services USDoHH. FAQs about NHII. [November 17, 2009]; Available from: http://aspe.hhs.gov/sp/NHII/FAQ.html.

[104] Irwin JY, Torres-Urquidy MH, Schleyer T, Monaco V. A preliminary model of work during initial examination and treatment planning appointments. British Dental Journal. 2009 Jan 10;206(1):E1; discussion 24-5.