Concept Extraction for Disability Insurance Payment Evaluation

by

Jeremy Lai

Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June, 2011

Author _____
Department of Electrical Engineering and Computer Science
May 25, 2011

Certified by _____
Peter Szolovits
Professor
Thesis Supervisor

Certified by _____
William J. Long
Principal Research Scientist
Thesis Supervisor

Accepted by _____
Dr. Christopher J. Terman
Chairman, Masters of Engineering Thesis Committee

Concept Extraction for Disability Insurance Payment Evaluation

by

Jeremy Lai

Submitted to the Department of Electrical Engineering and Computer Science
on May, 25, 2011, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Automated evaluation of claims for medical and disability insurance benefits poses a difficult challenge that will take years to be solved. The precise wording of insurance rules and the terse language in medical history files make it difficult for humans, let alone computers, to assess insurance payment qualification accurately. In this thesis, we work towards building a tool that will aid, but not replace, human evaluators. We automate the extraction of relevant parts of medical history files; if sufficiently accurate, this would eliminate the need for human evaluators to comb through hundreds of pages of medical history files. We first create a list of medical concepts, mainly disease and procedure names, from the cardiovascular section of the "Blue Book" for Disability Evaluation under Social Security. Then, using a variation of the longest common substring algorithm, we characterize each medical file line using its substring overlaps with the list of medical concepts. Finally, with human annotations of whether each medical file line is relevant or not, we build machine learning classifiers predicting each line's relevance using its overlap characterization. The classifiers we use are Naïve Bayes and Support Vector Machines.

Thesis Supervisor: Peter Szolovits
Title: Professor

Thesis Supervisor: William J. Long
Title: Principal Research Scientist

Acknowledgements

I would like to thank my advisor Peter Szolovits for guiding me through my research. I am also grateful for Bill Long, whose advice and encouragement were very helpful.

I would like to thank Arya Tafvizi and Kanak Kshetri for being such great cubicle mates. The coffee runs and midnight football throwing will be missed.

Finally, I would like to thank my family for all their care and support. I would not have survived the past five years without your love.

Contents

1 Introduction

A patient's medical history records contain a wealth of information that can be used for various purposes. Besides being an indicator for future medical risks and a guide to suitable procedures or medications for the patient, these medical history files provide a background story behind the patient's current condition. Should a patient choose to apply for medical insurance benefits, these records are a major factor in determining whether the patient's present condition qualifies. Insurance evaluators use these files as the first, and maybe most important, screening tool when evaluating applicants. We focus on the determination of eligibility for disability payments as defined by the U.S. Social Security system, and refer to the determination of whether the medical documents in a case qualify the claimant for benefits as "insurance evaluation."

Medical history files may be very long, reaching over a hundred pages of text, and for a human evaluator to comb through these files can be a tedious task. Moreover, when applicants disagree with the insurance evaluator's decision, they may choose to enter the long bureaucratic process of appealing the decision and even taking their cases to court. Thus, accurate evaluation is important, yet costly.

As with many other problems dealing with large amounts of natural text, insurance evaluation of medical files invites the use of natural language processing (NLP). In the past couple of decades, numerous research projects have been conducted on applying natural language processing to electronic medical records. Typical research objectives include medical concept extraction, assertion classification, de-identification, etc. Depending on the problem, different well-known tools of NLP appear in these studies, such as part-of-speech tagging or parsing, and often research involves solving traditional NLP problems, like named entity recognition or co-reference resolution, in the specific arena of medical records. Behind the application of NLP to medical records stands a set of methods that can be roughly divided between the relatively recent statistical learning methods, such as Support Vector Machines, and the older rule-based or dictionary approaches.

Our problem, insurance evaluation, is novel, interesting, and hard. It can be divided into three components: the extraction of relevant information from medical history files, the representation of insurance qualification rules, and the prediction of disability qualification under the representation of the insurance rules. In other words, an insurance evaluation system would first have to extract from the medical files relevant medical facts that may be used to judge

whether a patient qualifies for disability payments. Once relevant medical facts have been extracted, the system would have to have some computer representation of the insurance qualification rules that it could then apply to the extracted medical facts and determine a patient's disability qualification. Given the potential length of insurance rule books, the representation of the rules would ideally be automatically generated through applying NLP techniques to the rule books.

Automating each of the three tasks in insurance evaluation poses significant challenges. We have worked with a set of rules as codified by the Social Security Administration and have received access to the medical records of 77 claimants, all de-identified to protect their confidentiality. Of these, 27 were adjudicated as eligible for disability and 50 were denied.

In this thesis, we first describe the medical files and insurance rules with which we were provided. Then, we review the literature covering the three components of insurance evaluation and the relevance of existing methods to our dataset. Through this review we present why we focused our work on concept extraction instead of the other two components of insurance evaluation. We then describe the methods we used for concept extraction and their results. Specifically, we describe how we used a variation of the longest common substring algorithm to find all substring overlaps between a file line and a list of medical keywords. Using the author's annotations of the relevance of each line, we then trained machine learning classifiers to predict each line's relevance using the line's set of overlap substrings. The classifiers we used are Naïve Bayes and Support Vector Machines. After describing our methods of concept extraction, we describe how we tested our classifiers and how they performed in these tests. Finally, we discuss the implications of our results, shortcomings of our methods, and areas for future work.

2 Background

2.1 The Data and the Blue Book

The Social Security Administration (SSA) has dealt with insurance evaluation for several decades. The current rule book used by the Administration for disability evaluation, called the Blue Book, runs for 390 pages and covers all sorts of disabilities, from mental disorders such as schizophrenia to physical impairments such as hearing loss. To qualify for disability payments, applicants submit packets of materials pertaining to their claimed disability; among these packets is their medical history file detailing current and past medical problems, medications, procedures, and other relevant information. The medical history files provided to us by the SSA were from the Veterans Health Information Systems and Technology Architecture (VistA), the electronic health records system used by the Veterans Health Administration. In addition, we chose to study only cardiovascular disability cases because of our past experience in this area.

These files were electronic text files, de-identified and printed out from VistA. They ranged in length from less than twenty pages to over one hundred pages. Each file had section headers depending on what types of information the applicants wanted in their applications. The formatting of the files varied from section to section, and within each section the formatting also varied depending on how doctors took notes. In fact, some text formatting imitated physical forms, such as:

__X__ Symptom A _____ Symptom B __X__ Symptom C __X__ Symptom D

or

[x] Symptom A [ ] Symptom B [ ] Symptom C [x] Symptom D

For each file we were told whether the case was approved for disability or not, and if it was approved, which rule or sub-rule in the Blue Book the patient qualified under. Since we dealt with only cardiovascular cases, we only needed a thirty-six page subset of the Blue Book. Of these pages, only five pages were specific rules detailing the exact medical conditions and evidence needed for qualification; the first thirty-one pages elaborated on the SSA's definitions of different medical conditions. An example of these rules is shown in Appendix A. Though applicants submit other materials besides their medical history files when they apply for

disability, our partners at the SSA assured us that the justification for each approval case could be found in the medical files we were provided.

2.2 Different Problems in Insurance Evaluation

There are different sub-problems which contribute towards the aforementioned components of insurance evaluation; these include document segmentation, temporal reasoning, Blue Book rule extraction, and concept extraction. In this section we review some of the approaches to the first three sub-problems and present why the characteristics of our data set rendered these problems intractable or irrelevant.

Document segmentation allows us to segment the medical history files into tractable pieces, where each piece is about the same topic. There are several approaches to this problem that extend the classic document classification problem to within documents; they can be found in [3] and [13]. In our data, VistA divided each medical file into different sections, such as "All Problems," "Brief Demographics," and "Progress Notes"; the possible sections are given on page 59 of [18]. In addition, within the sections there were labeled subsections that could be extracted with simple string matching algorithms. Since most subsections were cohesive in nature, topical document segmentation was not pertinent to our medical files.

Because the rules of the Blue Book sometimes include time-related conditions, such as having a condition for at least three months, temporal reasoning may be seen as a useful sub-problem. Temporal reasoning is the extraction of temporal information, using this information to segment the text into temporal units, and then sorting or finding relationships between each temporal unit [5, 19].

With our data, we reverse-engineered VistA's print-out formats and used regular expressions to capture dates of most of the important events in each applicant's medical history. In addition, upon closer examination of the Blue Book rules, patients' medical events that were in the past would not be nearly as relevant as the current conditions of the patient; this obviated the need for a rigorous timeline of the patients' medical history. Furthermore, even though the narrative text components of medical histories would allow for temporal reasoning, most research on temporal reasoning in medical records reported in the literature has been designed for discharge summaries, which are only a small component of the narrative text within our data. Most of the narrative text in the medical histories was labeled "Progress Notes" that detailed

patients' medical condition at the time of visit. Thus, capturing VistA's time labels allowed us to come up with an accurate sorting of the relevant medical events in applicants' medical histories.

The third sub-problem, automated representation of the Blue Book rules, had few references in the literature. The most similar research attempt we could find was Popescu et al.'s natural language understanding system built to understand students' assertions and logic in geometry solutions [14]. We created a manual representation of the rules as a set of logical statements, but this representation contained clauses so specific that examples in the approved medical history files could not be found. This was partly due to our approval cases being predominantly (24 out of 27) related to three rules out of the eight rules. In addition, some rules were exceedingly complex, such as:

> 4.04 Part A: Sign- or symptom-limited exercise tolerance test demonstrating at least one of the following manifestations at a workload equivalent to 5 METs or less:
>
> 1. Horizontal or downsloping depression, in the absence of digitalis glycoside treatment or hypokalemia, of the ST segment of at least -0.10 milliovolts (-1.0 mm) in at least 3 consecutive complexes that are on a level baseline in any lead other than a VR, and depression of at least -0.10 millivolts lasting for at least 1 minute of recovery; or...

Despite our having a case classified as being approved under the above sub-rule, we could not find within the medical history file the evidence pertaining to the rule. Given these complications concerning the Blue Book rules, tackling the rules with NLP was not feasible. Instead, we chose to manually find the medical keywords that pertained to each rule. With this set of keywords, we found concept extraction the most applicable sub-problem of insurance evaluation.

2.3 Concept Extraction Background

Medical concept extraction has been studied extensively over the past decade. Its uses are many, such as creating and maintaining problem lists [10] or converting natural text clinical documents into a list of medical concept codes with modifiers [6]. Many of these attempts at medical concept extraction have focused on different ways of augmenting systems using extensive medical dictionaries, of which the most extensive is the Metathesaurus of the Unified Medical Language System (UMLS) created by the National Library of Medicine. The Metathesaurus combines medical terminology from many different sources, such as International Classification

of Diseases (ICD), Medical Subject Headings (MeSH), and Systematized Nomenclature of Medicine—Clinical Terms (SNOMED-CT), and provides a unified dictionary for looking up medical terminology. Having such a unified dictionary allows many medical synonyms to be categorized under the same entry, and makes the UMLS an indispensable resource for natural language processing in the medical domain.

MetaMap is a program built specifically for finding UMLS medical concepts. It uses a wide array of NLP tools, including shallow parsing, word sense disambiguation, and variant lookup, to aid the underlying procedure of looking up words in the Metathesaurus [1]. Various studies have been conducted regarding MetaMap's performance in concept extraction, usually simultaneously with assertion classification, which is the determination of whether a statement is positive, negative, or otherwise. Assertion classification will not be our focus, but techniques for assertion classification include NegEx [7] and Support Vector Machines [17].

Haug and Meystre compared MetaMap's performance with their system MPLUS2 in trying to create patient problem lists through extracting patient problems from free-text documents [10]. MPLUS2 uses Bayesian Networks with eleven semantic categories serving as nodes. A Bayesian model was then built to capture the Bayesian probabilities of transitioning from a specific concept in a one semantic category to another concept in a different semantic category. Using this, an optimal semantic categorization of a sentence could be calculated, and the words placed in the Problem category would be extracted as the patient's problem. MetaMap and MPLUS2's precision and recall were 0.775/0.398 and 0.693/0.402, respectively.

Denny et al. compared MetaMap with their system KnowledgeMap in extracting medical concepts from medical teaching documents describing lecture content [4]. KnowledgeMap uses several heuristic techniques designed with the setting, curricular documents, in mind. Some special characteristics of the documents included their outline format, the common use of abbreviations, and the fusion of medical concepts to create concepts not in the Metathesaurus. In this setting, MetaMap and KnowledgeMap had precision and recall performances of 0.85/0.78 and 0.89/0.82, respectively.

Like the creators of KnowledgeMap, Long used a heuristic approach towards concept extraction from discharge summaries that capitalized on formatting patterns within the given set of discharge summaries [9]. Once again, a heuristic approach tailored towards the dataset at hand out-performed MetaMap with recall measures of 0.96 and 0.876, respectively.

Another example of dataset specific processing is the use of punctuation processing, shorthand (acronyms, abbreviations, truncated words) expansion, and modifier/qualifier deletion by Travers and Haas in testing a concept extraction system with UMLS for nursing chief complaint (CC) forms from emergency departments [16]. Their research motivation was to help develop a more standardized terminology for CC forms, and to do so, they first tested how well the medical terminology used in CC forms mapped to a UMLS based system. They applied the text processing sequentially, allowing their concept extraction to go through several rounds of extraction with each round discovering concepts not found in previous rounds. Starting with 13,494 chief complaint entries, 5083 entries matched at least one UMLS concept, while the rest did not have any matches to UMLS concepts.

In addition to being used with and without MetaMap, UMLS can be used sometimes with other systems, such as MedLEE. MedLEE is a medical language processing system similar to MetaMap that can be configured with additional lexicons. In a study comparing MedLEE with its own lexicon and with only UMLS, a concept extraction system by Friedman et al. for discharge summaries performed with precision and recall of 0.93/0.81 and 0.96/0.60, respectively [5].

Just as in Haug's and in Friedman's study, UMLS-based concept extraction sometimes may not perform well with recall. That was our initial experience when applying Long's UMLS-based concept extraction and checking whether certain desired concepts had been found. We therefore chose a completely different approach, which we will discuss in the following chapter.

3 Methodology

Our approach to concept extraction revolved around string matching. For each medical file line, we automated the finding of all the different substrings that occurred both in the file line and in any of the key medical concepts we chose from the Blue Book. The overlapping substrings provided a characterization of each file line. Then, using human annotations of whether each line was relevant or not, we trained a Naïve Bayes classifier and Support Vector Machine to predict each line's relevance based on its set of string overlaps with the key medical concepts. In this chapter we will first discuss the motivation for our approach towards concept extraction. Then we will describe the preprocessing of the data set, our string matching algorithm, and finally our applications of Naïve Bayes and Support Vector Machines.

3.1 Motivation

Our literature review revealed that language processing heuristics tailored for the data set at hand allowed for better performance in concept extraction. Some of these heuristics included stemming (reducing a word to its root form), punctuation heuristics, and variant generation. One thing in common with the aforementioned heuristics is that oftentimes the medical text and the concept name have significant overlaps. For example, Haas et al. used punctuation heuristics to cope with text such as "dizzy/fever," or "congest" (congestion) [16]. In our text, "ankle/brachial indices" appeared with and without "index" or "indices" as:

ankle/brachial

ankle /brachial

ankle brachial

ankle/BK

In total, this one concept had up to eight different, but easily identifiable, variations in our dataset. This variety is not surprising considering that our set of medical history files originated from different hospitals and different departments within hospitals.

Just as punctuation and spacing introduce noise to concept extraction, different forms of words achieve the same effect. For example an "echocardiogram" can be referred to as "echocardiograph" or simply "echo." Currently, MetaMap and other systems approach this problem through listing the various forms a concept may appear in. Likewise, Porter's classic stemming algorithm uses a brute force approach through looking at all the common suffixes to

find core terms. Unfortunately, systems that rely on enumerated sets of terms fall short of complete coverage because of rare synonyms or an expanding terminology over time.

Unlike many of the concept extraction and assertion classification studies conducted in the past, in our data set, nearly all the medical history files came with a dated problem list listing the major medical difficulties facing the patient. Insurance evaluation requires a more nuanced and detailed understanding of the patient's condition than finding major medical conditions and assigning them a positive or negative classification. In the Blue Book, rules sometimes indicate that a condition must be had over a certain amount of time or that a certain symptom must be displayed under certain circumstances. The variety of logical conditions and the complexity of medical conditions in the Blue Book would require a system beyond the scope of our work. At the same time, concept extraction would be potentially quite useful for insurance evaluation.

Therefore, we approached concept extraction trying to mimic heuristic language processing through string matching and machine learning. Using a set of common substrings as inputs to a supervised machine learning algorithms, we hoped to test whether the machine learning algorithms would be able to extract the concepts successfully. The machine learning algorithm would basically be able to weed out those substrings that were unimportant and make decisions based on those that were important. Looking at our previous example, "echocardiogram," machine learning based on a overlap string input of "echocardio" would ideally note that "echocardiogram" and "echocardiograph" are similar while "echocardiogram" and "electrocardiogram" are not.

3.2 Pre-processing

Though our goal was that our string matching algorithm would be able to replace any form of rule-based processing of the data, for convenience, we processed the data to expunge irrelevant boiler plate sections of text. First, we eliminated all VistA print notes, which took up seven lines of text on every page. Second, we eliminated electronic signatures that were attached at the end of every sub-note of the medical history files. Finally, we sorted the sub-notes of each medical history file by date. This was done simply by noting the date of each sub-note, which was placed at each sub-note's beginning by VistA.

The most important task of pre-processing that we did was manually deciding which medical concepts were important in the Blue Book. We had considered trying to automate this

process, but as described before, we judged this to be infeasible. The keyword concepts we chose, attached in Appendix B, were from three out of the eight rules in the Blue Book, which are attached in Appendix A. We limited our study to these three rules because they accounted for the approval of the great majority of the approved cases (24 out of 27). Also, we chose not to study the 50 denied cases because we had no information from the SSA on why each case was denied. We left out one of the 24 remaining files because its file size, five times the second largest approved file's, was too large for annotating purposes. Thus, we operated on 23 approved medical history files using concepts from three Blue Book rules.

3.3 Longest Common Substring Algorithm and Application

The longest common substring is a well-known problem that can be solved in a variety of ways. For our purposes, we wanted to find all longest common substrings between two lines; these substrings would need to be of some minimal length so that trivial overlaps, such as overlaps of one letter, would not be returned. For example, if our inputs were "ankle/brachial pressure" and "His ankle has normal pressure," our outputs would be "ankle" and "al pressure" if our minimal length were set to five. In other words, our algorithm finds all substring overlaps of at least the minimal length set that are not adjacent to each other. If two substring overlaps were adjacent to one another, they were combined.

Our algorithm received as inputs a keyword string and a "main" string from the medical file. For each character in the main string, we first stored where it appeared in the keyword string. Then, for each following character in the main string, we checked whether the corresponding following characters in the keyword string were the same. Continuing this checking until a different character was found, the algorithm stored the longest overlap starting with the main character at hand and skipped to the next character in the main string that followed the overlap.

Our longest common string algorithm was used to find the overlaps between each line and our list of keywords. The choice of one line of text may seem arbitrary, but the formatting of the documents was such that lines were often atomic units. We considered using the Stanford Parser to break our text into sentences, but the intermix of sentences and non-sentences in the files forced the parser to perform poorly.

Another parameter in our algorithm was the minimum length of overlaps, which we set to five characters. This was because the shortest keyword had five characters, and setting the minimum length any shorter would only serve to increase the amount of noise returned by the system. Since spaces counted as a character, our minimum length was in reality only four alphabet characters if the overlap were at the beginning or the end of a word, and five only in the middle of words.

In all, there were a total of 22,462 lines with some overlap of at least five characters with the keywords. We proceeded to annotate these lines to determine whether the line was actually relevant to any keywords. Of these, 1,934 lines were annotated as positive or an average of 80.58 relevant lines per medical history file. This number is actually an underestimate of the number of relevant lines because two lines such as the following would only be labeled as one relevant line (the first line):

Edema:

No

The lines annotated as positive were thus signals of relevant areas in the medical files.

We used the set of all overlap strings to be a set of features that could have value of 1 or 0, denoting whether a file line had contained that overlap string. Each line thus had values of 1 for the overlap strings found in it and values of 0 for the overlap strings not found it (but found elsewhere). Two examples in our data set are:

1. "evidence an acute compression fracture. minimal hypertrophic"
   Classification: -1 (not relevant)
   Overlaps: (keyword, overlap)
   systolic pressure , press
   radionuclide perfusion scans , sion
   ankle systolic pressure , press
   toe systolic pressure , press

which is equivalent to

Class: -1

Features: "press" 1; "sion" 1; else 0

and

2. "venous insufficiency with left ankle ulcers s/p vascular surgery and skin"

Classification: 1 (relevant)

Overlaps: (keyword, overlap)

revascularization , vascular

venous insufficiency , venous insufficiency

ulceration , ulcer

ankle/brachial , ankle

ankle systolic pressure , ankle

which is equivalent to

Class: 1

Features: "vascular" 1; "venous insufficiency" 1; "ulcer" 1; "ankle" 1; else 0

As can be seen from our examples, certain overlaps (e.g. "press" and "sion") would ideally be distinguished through a machine leaning classifier to be worthless in prediction, while others (e.g. "venous insufficiency") would be judged as useful. With an understanding of these assignments and annotations, we may proceed to discuss the classifiers we implemented.

3.4 Naïve Bayes

A Naïve Bayes classifier is a based on Bayes law:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A)\,P(A)}{P(B)}$$

An extension of this is

$$P(A|B \cap C) = \frac{P(B \cap C \mid A)P(A)}{P(B \cap C)} = \frac{P(B|A)P(C|A)P(A)}{P(B)P(C)}$$

assuming independence between events B and C and also conditional independence on A. Suppose event A has two options; then, by calculating the probabilities of each event given the other events, we can make a classification based on which probability is higher. In our case we can consider event A as the classification of a line and events B, C, etc. as whether each of the possible strings were found to overlap between a line and the keywords. If we then assume the overlap strings are independent of each other and that they are conditionally independent based on the line's relevance, then our problem becomes one of estimating $P(X|A)$, $P(X)$, and $P(A)$ in the second version of Bayes formula. Given our data set, we can make that estimate by calculating the statistics of appearances of each overlap string and whether the overlap string was

associated with a positive or negative classification of the line it was in. The formulas for estimation are as follows:

$$P(A) = \frac{\#\ of\ positive/negative}{\#\ of\ lines\ total}$$

$$P(X) = \frac{\#\ of\ overlap\ appearances\ total}{\#\ of\ lines\ total}$$

$$P(X|A) = \frac{\#\ of\ appearances\ of\ X\ and\ positive/negative}{\#\ of\ appearances\ of\ positive/negative}$$

Given the data set, these statistics can be calculated in a short amount of time. To avoid estimating zero statistics for rare events that happened not to be seen in the training data, we employ Laplace smoothing, which adds a small constant alpha to the counts from which conditional probabilities are derived. Because some of these probabilities are quite small, we use the logarithms of the probabilities instead, which leads to an additive form of Bayes rule.

Once these probabilities have been calculated, we have a Bayesian model of the relationship between the overlap strings and the classification of relevance. Note that our Bayesian model is reminiscent of the oft-seen n-gram models in natural language processing.

Armed with the smoothed Bayesian model, we can then predict the classification of a line given its overlap strings. We look up the relevant probabilities in our Bayesian model, and compute the logarithm of the probability that the line should be classified as positive versus negative. Depending on which probability is larger, we can choose the classification of the line.

In our use of Naïve Bayes classifiers, we also engaged in a simple tuning procedure based on our assumption that for our scenario recall is more important than precision. This tuning can be done by choosing to classify a line as positive if

$$logP(positive) + c > logP(negative)$$

We chose to tune our classifier so that the number of true positives it returned was similar to the number returned by our Support Vector Machine classifier, which we describe next.

3.5 Support Vector Machines

Support Vector Machines (SVM) are another common tool for classification problems. They have been used with success for natural language processing such as in [15] and [17]. SVM's transform a classification problem into an optimization problem [8]. Given a set of points in an

n-dimensional space, SVM's use a kernel function to map the points to higher dimensional space in an attempt to make the points linearly separable. An objective cost function is optimized so that the distance from the "line" to the points is maximized given the conditions that each class of points lie on different sides of the line. A perfect classification is usually impossible, so an error term can be incorporated in both the conditions and the cost function.

For our study, we used the popular LIBSVM implementation of SVM's. Following their advice on optimizing the parameters for their SVM, we ended up using a linear kernel. Also, given that we emphasized recall over precision, we weighted the errors of false negatives three times as much as those of false positives.

4 Testing and Results

4.1 String Matching Results

When run on all 23 files, our common substring algorithm yielded 472 different substrings and 55,236 appearances across all the substrings. The most common substring was "ation " with 17,197 appearances; the second most common, "active," only appeared 2,192 times. The median number of appearances was twelve times, and the average, not including "ation," was 80.76 times.

A quick scan over the substrings with the most appearances reveals many common cardiovascular or medical terms, such as " disease," " artery," "cardio," "heart," etc. Some suffixes common in this domain also reveal themselves, such as "ation ," "eral ," and "sion ." On the other end of the frequency spectrum, some interesting substrings include "venous insuff" which appeared once, "venous insuffi" which appeared twice, and "echocardiogra" which appeared six times. Each of the lines with these three terms was annotated as relevant.

To test whether our substring characterizations of each file line was useful in predicting relevance, we used Leave One Out Cross Validation, which we shall describe in the next section.

4.2 Cross Validation

For comparison, we also ran a rule-based classifier that simply looks for each of the keywords literally to appear in the case text. We then assume that any occurrence of such a keyword implies relevance. To test our system, we chose to conduct Leave One Out Cross Validation. For each file, we trained the Bayes and SVM classifiers on the remaining 22 case files and tested it on the selected file. We report results averaged over the 23 trials. We measured the results through precision, recall, F_1-measure, and F_2-measure as defined as follows:

$$Precision = \frac{\#\ True\ Positives}{\#\ True\ Positives + \#\ False\ Positives}$$

$$Recall = \frac{\#\ True\ Positives}{\#\ True\ Positives + \#\ False\ Negatives}$$

$$F_1 - measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

$$F_2 - measure = \frac{5 \times Precision \times Recall}{4 \times Precision + Recall}$$

Precision, recall, and the F_1-measure are standard ways of evaluating the performance of a binary classifier. The F_2-measure weights recall twice as heavy as precision. We use the F_2-measure because our insurance evaluation scenario needs the extraction of concepts more than it is harmed by false positives. The results of the above statistical measures are shown in the figure below:

	Precision	Recall	F_1-Measure	F_2-Measure
Rules	98.26%	61.43%	75.60%	66.41%
Bayes	42.77%	89.25%	57.82%	73.31%
SVM	84.55%	88.00%	86.24%	87.29%

Figure 1: Cross Validation Results

The rule-based baseline classifier performed nearly perfectly in precision because when the keyword string was matched, the line was almost always relevant. The only exceptions were because of ambiguities in common use of medical terms; for example, "ulcer" in this section of the Blue Book rules normally refers to a venous ulcer, but was also found in the context of "mouth ulcer," which was irrelevant to the rule. Recall was relatively weak, however, because without common substring matching, variants of the keyword were not identified even though the human annotator had deemed them to be relevant. For example, phrases such as "echocardiograph" or "... echo results ..." were annotated as relevant because of the keyword "echocardiogram."

The SVM classifier was clearly the best classifier by greatly out-performing the Bayes classifier in precision and the rule-based classifier in recall. Both F-measures reflect SVM's relatively outstanding performance. The Bayes classifier was arguably better than the rule-based classifier; this is only true because we emphasize the importance of recall over precision. Because the threshold of the Bayes classifier was tuned to yield recall similar to that of the SVM, those two do in fact show very similar recall. However, at that level of recall, the SVM is far more accurate in its precision than Bayes, which finds many more false positives.

5 Discussion

In our thesis we had two goals: determine the feasibility of different sub-problems of insurance evaluation and develop solutions to any important and tractable sub-problems. In Section 2.2, we presented the case that the most relevant and feasible sub-problem is concept extraction. Here we will discuss the implications, short-comings, and future possibilities of our work on concept extraction from medical history files.

5.1 Implications

Our original goal was to be able to find all medically relevant lines in a file with regards to a given set of rules. Our system does this in part: it find those lines which are relevant to a set of keywords, but it does not have the semantic capabilities that would allow it to find all relevant lines regardless of whether it contains a substring overlap with the keywords. However, our research motivation was not to replace the semantic capabilities of UMLS, but the heuristic language processing used to process text before looking words up in the UMLS. When evaluated as such, a system to help identify all possible string variations of a word, our SVM classifier performed well. The SVM classifier has the reasonably high precision and recall of 0.8455 and 0.88, respectively, in trying to predict what lines had the original form of a keyword phrase. As we had suspected, the classifier did discover which character overlaps were important and which were unimportant when judging overlaps between file lines and keywords.

5.2 Work Critique

Our work presents several issues, both in methodology and in practicality, which we shall briefly address.

First, though our classifier was essentially able to achieve the same purposes as heuristics for language processing and dictionary look-ups of variants or suffixes, we question the time to reward ratio. As the author personally annotated the data, it became clear that the amount of time spent annotating the data was equivalent to going through the lines of the medical files and recording all the variants of our set of keywords. Though there may be some intellectual satisfaction in having a classifier that has no need for a dictionary of variants, the amount of time spent in annotating data as opposed to constructing a dictionary are on par with each other.

Second, like many supervised machine learning studies, more data would have made our study more robust. In addition to using the approved medical files for our study, using denied cases and cases that were approved under different rules would have been helpful in discovering whether the higher occurrence of relevant lines had any effect on our system's performance. Also, to limit the number of lines pre-selected as possibly relevant, we chose a minimum substring overlap of five characters. With a different set of rules, which may have shorter words than five characters, our study results become less applicable.

Another weakness in our system is the limited number of keywords fed in to the system. Without further study, it is impossible to conclude whether or not this system would be able to handle a significantly larger number of keywords, such as would arise from broadening its medical focus to many other causes of disability.

5.3 Future Work

There are several manners in which we can continue our work both in concept extraction and in the larger picture of insurance evaluation. First, expert annotation, more keywords from rules, and more features other than substring matching should all contribute towards better concept extraction. Once relevant lines have been found, an algorithm for connecting these sometimes scattered relevant lines into a coherent re-presentation of the medical history file would provide a more user-friendly version of concept extraction for insurance evaluators.

To continue the work on insurance evaluation in general, more data that covered a wider variety of Blue Book rules would be useful. Currently, a few simple heuristics (edema implies approval under Rule 11, claudication implies Rule 12, else Rule 4) can predict for most files which approval rule it was approved under. In addition, more information on why each file was approved would be quite helpful in trying to see whether we can recapture the decision making process of insurance evaluators. Considering the limited information of this kind available, asking insurance evaluators to annotate the medical history files for key areas or key lines that contributed to an approval or denial process would open the possibility of trying to conduct a machine learning study of predicting these important areas.

6 Closing Remarks

In our thesis, we explored the different components of insurance evaluation. We argued that the structure of our medical files and the Blue Book made concept extraction the most important, yet feasible, task. We developed an algorithm to reduce each file line to its substring overlaps with a list of medical concepts. We showed that using SVM's, we could make reasonably accurate predictions, as determined by a human annotator, of the relevance of each line in the medical files.

A recent Wall Street Journal article noted that the Social Security Administration has a backlog of 730,000 cases that still need evaluation [12]. This backlog is much more than a theoretical problem where natural language processing may be useful; it is a roadblock standing between hundreds of thousands of people with disabilities and the financial support they desperately need. We hope that our work will be a small step in helping the SSA make fair and speedy disability insurance evaluations.

Appendix A

Blue Book Rules 4, 11, and 12

4.04 Ischemic heart disease, with symptoms due to myocardial ischemia, as described in 4.00E3-4.00E7, while on a regimen of prescribed treatment (see 4.00B3 if there is no regimen of prescribed treatment), with one of the following:

A. Sign- or symptom-limited exercise tolerance test demonstrating at least one of the following manifestations at a workload equivalent to 5 METs or less:

1. Horizontal or downsloping depression, in the absence of digitalis glycoside treatment or hypokalemia, of the ST segment of at least -0.10 millivolts (-1.0 mm) in at least 3 consecutive complexes that are on a level baseline in any lead other than a VR, and depression of at least -0.10 millivolts lasting for at least 1 minute of recovery; or

2. At least 0.1 millivolt (1 mm) ST elevation above resting baseline in non-infarct leads during both exercise and 1 or more minutes of recovery; or

3. Decrease of 10 mm Hg or more in systolic pressure below the baseline blood pressure or the preceding systolic pressure measured during exercise (see 4.00E9e) due to left ventricular dysfunction, despite an increase in workload; or

4. Documented ischemia at an exercise level equivalent to 5 METs or less on appropriate medically acceptable imaging, such as radionuclide perfusion scans or stress echocardiography.

OR

B. Three separate ischemic episodes, each requiring revascularization or not amenable to revascularization (see 4.00E9f), within a consecutive 12-month period (see 4.00A3e).

OR

C. Coronary artery disease, demonstrated by angiography (obtained independent of Social Security disability evaluation) or other appropriate medically acceptable imaging, and in the absence of a timely exercise tolerance test or a timely normal drug-induced stress test, an MC, preferably one experienced in the care of patients with cardiovascular disease, has concluded that performance of exercise tolerance testing would present a significant risk to the individual, with both 1 and 2:

1. Angiographic evidence showing:

a. 50 percent or more narrowing of a nonbypassed left main coronary artery; or

b. 70 percent or more narrowing of another nonbypassed coronary artery; or

c. 50 percent or more narrowing involving a long (greater than 1 cm) segment of a nonbypassed coronary artery; or

d. 50 percent or more narrowing of at least two nonbypassed coronary arteries; or

e. 70 percent or more narrowing of a bypass graft vessel; and

2. Resulting in very serious limitations in the ability to independently initiate, sustain, or complete activities of daily living.

*4.11 **Chronic venous insufficiency*** of a lower extremity with incompetency or obstruction of the deep venous system and one of the following:

A. Extensive brawny edema (see 4.00G3) involving at least two-thirds of the leg between the ankle and knee or the distal one-third of the lower extremity between the ankle and hip.

OR

B. Superficial varicosities, stasis dermatitis, and either recurrent ulceration or persistent ulceration that has not healed following at least 3 months of prescribed treatment.

*4.12 **Peripheral arterial disease***, as determined by appropriate medically acceptable imaging (see 4.00A3d, 4.00G2, 4.00G5, and 4.00G6), causing intermittent claudication (see 4.00G1) and one of the following:

A. Resting ankle/brachial systolic blood pressure ratio of less than 0.50.

OR

B. Decrease in systolic blood pressure at the ankle on exercise (see 4.00G7a and 4.00C16-4.00C17) of 50 percent or more of pre-exercise level and requiring 10 minutes or more to return to pre-exercise level.

OR

C. Resting toe systolic pressure of less than 30 mm Hg (see 4.00G7c and 4.00G8).

OR

D. Resting toe/brachial systolic blood pressure ratio of less than 0.40 (see 4.00G7c).

Appendix B

Keywords from Rules

Rule 04:

ischemic heart disease
myocardial ischemia
exercise tolerance test
ST segment
systolic pressure
ischemia
radionuclide perfusion scans
stress echocardiography
revascularization
coronary artery disease
angiography
nonbypassed coronary artery
bypass graft vessel
activities daily living

Rule 11:

venous insufficiency
edema
superficial varicosities
stasis dermatitis
ulceration

Rule 12:

peripheral arterial disease
claudication
ankle/brachial
ankle systolic pressure
toe systolic pressure
toe/brachial

Bibliography

[1] A. Aronson, F. Lang. An overview of MetaMap: historical perspective and reent advances. *Journal of American Medical Informatics Association*. 2010;17:229-236.

[2] P. Bramsen, P. Deshpande, Y. Lee, and R. Barzilay. Inducing Temporal Graphs. Proceedings of EMNLP-06, 2006.

[3] J. Canny, T. Rattenbury. A Dynamic Topic Model for Document Segmentation. Technical Report No. UCB/EECS-2006-161.

[4] J. Denny, J. Smithers, R. Miller, A. Spickard. "Understanding" Medical School Content Using KnowledgeMap. *Journal of the American Medical Informatics Association*. 2003;10:351-362.

[5] C. Friedman, H. Liu, L. Shagina, S. Johnson, G. Hripcsak. Evaluating the UMLS as a Source of Lexical Knowledge for Medical Language Processing. Proc. AMIA Symp. 2001;189-193.

[6] C. Friedman, L. Shagina, Y. Lussier, G. Hripcsak. Automated Encoding of Clinical Documents Based on Natural Language Processing. *Journal of the American Medical Informatics Association*. 2004;11:392-402.

[7] H. Harkema, J. Dowling, T. Thornblade, W. Chapman. ConText: An algorithm for determining negation, experiencer, and temporal status from clinical reports. *Journal of Biomedical Informatics*. 2008;42:839-851.

[8] C. Hsu, C. Chang, C. Lin. A Practical Guide to Support Vector Classification. Department of Computer Science, National Taiwan University. April, 2010.

[9] W. Long. Lessons Extracting Diseases from Discharge Summaries. Proc AMIA Symp. 2007; 478-482.

[10] S. Meystre, P. Haug. Comparing Natural Language Processing Tools to Extract Medical Problems from Narrative Text. Proc AMIA Symp. 2005; 525-529.

[11] S. Meystre, P. Haug. Natural language processing to extract medical problems from electronic clinical documents: Performance evaluation. *Journal of Biomedical Informatics*. 2003;39:589-599.

[12] D. Paletta. Disability-Claim Judge Has Trouble Saying 'No'. Wall Street Journal on the Web. 19 May, 2011. URL: http://online.wsj.com/article/SB10001424052748704681904576319163 605918524.html. [Accessed 24 May, 2011].

[13] J. Ponte, W. Croft. Text Segmentation by Topic. Computer Science Department, University of Massachusetts, Amherst.

[14] O. Popescu, V. Aleven, K. Koedinger. Logic-Based Natural Language Understanding for Cognitive Tutors. *Natural Language Engineering.* 2005; 1:1-15.

[15] T. Sibanda. Was the Patient Cured? Understanding Semantic Categories and Their Relationships in Patient Records. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology. May 2006.

[16] D. Travers, S. Haas. Using nurses' natural language entries to build a concept-oriented terminology for patients' chief complaints in the emergency department. *Journal of Biomedical Informatics.* 2003;36:260-270.

[17] O. Uzuner, X. Zhang, T. Sibanda. Machine Learning and Rule-based Approaches to Assertion Classification. *Journal of the American Medical Informatics Association.* 2009;16:109-115.

[18] VistA Health Summary Technical Manual. Version 2.7. Feb, 2002.

[19] L. Zhou, G. Hripcsak. Temporal Reasoning with Medical Data—A review with emphasis on medical natural language processing. *Journal of Biomedical Informatics.* 2006;40:183-202.