University of Wisconsin Milwaukee
UWM Digital Commons

Theses and Dissertations

December 2014

Text Mining of Patient Demographics and Diagnoses from Psychiatric Assessments

Eric James Klosterman
University of Wisconsin-Milwaukee

Recommended Citation

Klosterman, Eric James, "Text Mining of Patient Demographics and Diagnoses from Psychiatric Assessments" (2014). *Theses and Dissertations*. 613.
https://dc.uwm.edu/etd/613

TEXT MINING OF PATIENT DEMOGRAPHICS AND DIAGNOSES

FROM PSYCHIATRIC ASSESSMENTS

by

Eric Klosterman

A Thesis Submitted in

Partial Fulfillment of the

Requirements of the Degree of

Master of Science

in Health Care Informatics

at

The University of Wisconsin-Milwaukee

December 2014

ABSTRACT

TEXT MINING OF PATIENT DEMOGRAPHICS AND DIAGNOSES FROM PSYCHIATRIC ASSESSMENTS

by

Eric Klosterman

The University of Wisconsin-Milwaukee, 2014
Under the Supervision of Professor Rashmi Prasad

Automatic extraction of patient demographics and psychiatric diagnoses from clinical notes allows for the collection of patient data on a large scale. This data could be used for a variety of research purposes including outcomes studies or developing clinical trials. However, current research has not yet discussed the automatic extraction of demographics and psychiatric diagnoses in detail. The aim of this study is to apply text mining to extract patient demographics – age, gender, marital status, education level, and admission diagnoses from the psychiatric assessments at a mental health hospital and also assign codes to each category. Gender is coded as either Male or Female, marital status is coded as either Single, Married, Divorced, or Widowed, and education level can be coded starting with Some High School through Graduate Degree (PhD/JD/MD etc. Level). Classifications for diagnoses are based on the DSM-IV. For each category, a rule-based approach was developed utilizing keyword-based regular expressions as well as constituency trees and typed dependencies. We employ a two-step approach that first maximizes recall through the development of keyword-based patterns and if necessary, maximizes precision by using NLP-based rules to handle the problem of ambiguity. To develop and evaluate our method, we annotated a corpus of 200 assessments, using a portion of the corpus for developing the method and the rest as a test set. F-score was satisfactory for each category (Age: 0.997; Gender: 0.989; Primary Diagnosis: 0.983; Marital Status: 0.875; Education Level: 0.851) as was coding accuracy (Age: 1.0; Gender: 0.989; Primary Diagnosis: 0.922; Marital Status: 0.889; Education Level: 0.778). These results indicate that a rule-based approach could be considered for extracting these types of information in the psychiatric field. At the same time, the results showed a drop in performance from the development set to the test set, which is partly due to the need for more generality in the rules developed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would first like to thank my advisor, Rashmi Prasad, for her expertise and guidance throughout this study. I appreciated her high expectations as well as her patience as I slowly developed my current knowledge and understanding of natural language processing techniques. I also am thankful for her allowing me the opportunity to apply these techniques in the psychiatric field, where I think natural language processing can provide a great benefit but has not yet been explored in depth. I would also like to thank Dr. Pamela Bean for her assistance with obtaining permission to use the corpus used in this study, as well as for her words of encouragement and wisdom as I embarked on this journey. Additional thanks to Rohit Kate for introducing me to Python, which greatly helped me while completing this study, as well as thanks to Timothy Patrick for inspiring me to pursue my project idea as this current thesis.

Chapter 1: Introduction

1.1 Summary of the Study

Text mining involves the use of computer programs to systematically search through large amounts of text documents in order to extract relevant information. The objective of this study involves the use of text mining techniques to automatically extract clinically relevant data from the semi-structured and unstructured text portion of clinical documents. For this study, text mining will be used on a corpus of admission psychiatric assessments of patients from a mental health hospital. The objective is to automatically extract *patient demographics* and *admission diagnosis* information that are currently being manually collected by staff members at that hospital for the purpose of outcomes studies and research. The targeted demographics in each document consist of the patient's *age*, *gender*, *marital status*, and *education level*, each of which are also given a *code* for storage in a database. The corpus used in this study was approved by the Human Subjects Committee at Rogers Memorial Hospital and this study was also approved by the Institutional Review Board at UW-Milwaukee.

Approaches in text mining diagnoses from clinical texts have been extensively discussed, but only in the medical domain. Extraction of a wide variety of psychiatric diagnoses is yet to be explored. Prior research literature in extracting demographics from any text, clinical or otherwise, is also very limited and previous approaches have not been discussed in detail. Additionally, several past demographics extraction methods have utilized machine learning but rule-based methods were not considered.

The goal of this study is to determine if using rule-based text mining algorithms to automate the process of extracting patients' admission psychiatric diagnoses and the aforementioned demographics is feasible.

Research questions include:

1. Can the algorithms identify the appropriate text to determine the correct code for each category?
2. Are keyword-based pattern matching rules sufficient to accomplish this task, or are additional rules using natural language processing based on deeper syntactic and semantic processing necessary?
3. Are any text mining techniques used for one particular type of information also applicable to other kinds of information (i.e. is an algorithm generalizable)?

Because the corpus of admission psychiatric assessments consists of both structured and unstructured sections, some text mining tasks are anticipated to be more complex than others. Age and gender are presented in a consistent format throughout the corpus, therefore their respective extraction methods should be straightforward. Diagnoses are also available in a structured format but will require additional processing in order to exclude diagnoses that are inconclusive or currently in remission. Marital status and education level are both exclusively found in the narrative sections of the assessment and will require the most sophisticated approaches involving semantic processing and syntactic relations in order to identify the relevant text and infer the correct codes.

Separate algorithms were developed for each category using 110 documents as the development set and another 90 documents as the test set. The size of the corpus is small due to permission requirements from the hospital to utilize the original assessments from the electronic health record; these documents contained protected health information and had to be manually de-identified. The same documents were included in the development set and the test set for each algorithm. Each algorithm was either manually or automatically evaluated against a gold standard annotated version of the corpus. All algorithms were first developed using only keyword-based pattern matching rules for text detection and coding, following which errors were evaluated to determine whether they could be resolved using additional NLP-based rules.

NLP-based rules, where explored to improve the performance of the algorithm, were in turn developed over the development set. These rules utilized both constituent trees and typed dependencies from the Stanford Parser. Final evaluation was performed over the test set as an ablation study, where each NLP-based rule was successively removed to demonstrate the impact of that rule on the performance. For both the keyword-only version of the algorithm and the NLP-based version, we present and compare the performance of the algorithm over the development set as well as the test set.

For all categories, i.e., the admission diagnosis and the different types of demographics, the final versions of the algorithms produced codes with a high degree of accuracy. All algorithms also showed acceptable text accuracy with regard to identifying the correct source text as the basis for the corresponding code. F-scores indicating the algorithms' ability to identify all relevant text were also satisfactory. The algorithm for extracting age was the most accurate (code and text accuracy=1.0) and had the highest F-score overall (0.997). Gender had the second highest code accuracy (0.989), text accuracy (0.989), and F-score (0.989). Of the first five admission diagnoses extracted from each assessment, the 5th diagnosis in each assessment was the most accurately coded (0.944) but the 1st (or primary) diagnosis in each assessment was the most accurate in terms of text accuracy (0.978) and F-score (0.983). Performance of the algorithm used to extract marital status was also satisfactory (code accuracy = 0.889, text accuracy = 0.878, F-score = 0.875) as was performance of the algorithm used to extract education level (code accuracy = 0.778, text accuracy = 0.856, F-score = 0.851).

Education level was the most complex category and required additional NLP-based rules to supplement the keyword-based pattern matching rules. Marital status, while not as complex as education level, also needed additional NLP-based rules. Diagnosis, age and gender categories were identifiable with high accuracy with only keyword-based matching and did not require any NLP-based rules. The most generalizable component used in the algorithms was an exclusion rule that detects sentences whose subjects are not related to the patient, which led to considerable improvement in accuracy for both marital status and education level. The results of this study indicate that an automatic text mining approach could be developed to accomplish these tasks at an acceptable level of accuracy.

1.2 Organization of the Thesis

This thesis is divided into seven chapters. After this first Introduction chapter, the second chapter features a review of past related research. The third chapter discusses general methods such as development of the annotation schema used for this study as well as the development of the general purpose algorithms that were used to parse the sections of the psychiatric assessments and automatically evaluate the text mining algorithms' output. The fourth chapter reports the development and results of the algorithm for automatically extracting education level from the corpus, and the fifth chapter reports the same for the algorithm extracting marital status. The sixth chapter discusses the development and the results of three text mining algorithms: diagnosis, age, and gender extraction. The seventh and final chapter features general discussion and conclusions regarding the entire study as a whole.

Chapter 2: Background/Literature Review

The current body of literature available in text mining research involving extracting demographics from texts is scarce. It is often not discussed in as much detail in comparison to work in extracting diagnoses, which can be considered a higher priority due to the need for diagnostic information for a variety of purposes such as billing, analyzing patient outcomes, and clinical research. The existence of prior text mining in demographics however does illustrate an interest in the information extraction domain in systems that can collect such information as well. This chapter will review the available literature related to these extraction tasks and discuss how it is related to the present study's research focus. This chapter will also include a review of methods in rule-based natural language processing (NLP) using negation detection and dependency parsing, both of which will be explored during this study.

2.1 Text Mining of Clinical Diagnoses

The extraction of medical problems and diagnoses has already been explored with considerable success. For example, one study compared the performance of keyword-based text matching and NLP-based approaches for extracting medical problems from free-text clinical evaluations and documents. The results improved with the addition of NLP (keyword F-score of 0.61 vs. highest F-score of 0.86 with NLP) [1]. This keyword vs NLP comparison approach will also be utilized in the present study to show the effects of keyword-based pattern matching in conjunction with NLP-based rules on extracting information, although it is anticipated that it will only be necessary for demographics and not diagnoses due to diagnostic information being available in a semi-structured format in our corpus rather than in free-text.

Other studies have utilized MetaMap, which recognizes concepts found in the UMLS Metathesaurus, to extract medical problems. One study used it in conjunction with NegEx [2] for negation detection. Performance was satisfactory when using the complete default UMLS Metathesaurus data set (recall = 0.74, precision = 0.756). However, recall increased (0.896) when a custom subset was created focusing only on the medical problems that were related to the study's research goals with a non-significant decrease in precision [3]. Disorders matched in MetaMap have also been used as part of a feature set in training a system using Conditional Random Fields to extract disorders from clinical text [4]. The creation of a subset of diagnoses will also be used in the present study, however psychiatric diagnoses using the DSM-IV vocabulary in the Metathesaurus will be used rather than medical diagnoses. A rule-based negation detection approach will also be used similar to NegEx, however its negation rules will be designed in alignment with the research needs of the psychiatric hospital at which this study is taking place.

Additionally, an application known as HITEx (Health Information Extraction tool) has been used to extract principal diagnoses from discharge summaries with a focus on asthma and COPD patients. HITEx consists of a series of open-source modules that could be arranged into a pipeline depending on the extraction task. Examples of available modules included a section splitter, section filter, sentence splitter, negation finder, and UMLS concept mapper. It achieved an accuracy of 82% [5]. This modular approach will also be used in the present study, which aims to develop algorithms that are general enough to be used for different extraction tasks.

Although these studies demonstrate the effectiveness of extracting medical diagnoses, there hasn’t been to date a report of extracting a wide variety of psychiatric diagnoses from clinical text. Not all of the aforementioned studies explored the task of coding diagnoses or mapping them to controlled terminologies after extraction either. As a result this present study hopes to contribute to the more overlooked application of text mining in in the psychiatry domain by extracting and coding diagnoses that are related to mental health.

2.2 Text Mining of Psychiatric Diagnoses

Although little has been reported in terms of extracting psychiatric diagnoses, there have been several reported uses of natural language processing techniques to assign a particular psychiatric diagnosis classification to patients using narrative clinical notes. For example, one study classified patients as having clinician-diagnosed Binge Eating Disorder (BED) using a rule-based approach with several iterations of development. The final method achieved a classification accuracy of 91.8% accuracy and a sensitivity of 96.2%. Validation metrics during development included evaluating whether the relevant text was correctly identified as being relevant to BED in addition to whether the final classification of either having or not having BED was correct [6]. Similar metrics will be used in the present study to determine classification accuracy and that the classification was derived from the correct text.

Another study had the goal of classifying patients with depressive disorders based on clinical notes. A logistic regression classifier was used to determine whether patients were “depressed” or “well” during a particular visit based on the presence or absence of certain terms. Regular expressions were used to identify keywords and additional negation and context algorithms were applied to increase precision. Performance of the system was significantly more accurate based on the area under the receiver operating characteristic curve (0.85-0.88) in comparison to classifying depressive states using only ICD-9 codes (0.54-0.55). The results of this classification task was then used to determine longitudinal patient outcomes in terms of whether the patient was responsive or resistant to treatment [7].

NLP has also been used to classify trauma survivors as having either high or low risk for developing Post-Traumatic Stress Disorder (PTSD) based on their self-narratives. A set of keywords were selected based on their significant frequency in either the high or low risk categories based on a chi-square test. The keywords were then used to train a classifier. This approach was able to perform the classification task with 85% sensitivity and 78% specificity. When unigrams, bigrams, and trigrams were added as features in order to incorporate the relationship between consecutive words, results using three different machine learning approaches were compared. The most successful approach overall was with a product score model as a classifier using only unigrams as features and was comparable in performance to the keyword-only approach [8].

Although these studies were successful in identifying whether someone had a single particular psychiatric diagnosis or not, each of the approaches in these studies are limited in the scope of diagnoses that they can identify. They also rely on lexical items found in free-text that only indicate a high probability of the patient having a particular diagnosis, and do not aim to determine what the clinician's actual diagnosis was. Although these approaches could potentially be helpful in a decision support application, they do not generalize to a wide variety of diagnoses that could be coded and stored in a repository. The present study aims to develop an approach that can identify the majority of common mental health disorders that a given patient has been diagnosed with by the clinician in a psychiatric assessment.

2.3 Text Mining of Demographics

Existing literature has discussed automatically extracting general demographics from clinical research articles. One approach extracted demographics of subjects in structured and unstructured reports of randomized clinical trials by only focusing on sentences in the Methods section. This attributed to the high performance of the approach (F-score of 91%) which used text classification and a Hidden Markov Model [9]. Another study extracted general demographics using a mark-up tag set with a supervised machine learning approach in descriptions of clinical case studies. Although a high precision was achieved (91.6%), recall could have been improved (73.1%) [10]. These reports only discuss extracting demographics very broadly however and do not compare results between different types of demographics in detail.

Applications used to extract certain demographics on a large scale are also currently in use in hospitals. For example, LifeCode has been produced for diagnostic radiology and can extract certain demographics such as age and gender from free-text documents. It also can search for patient diagnoses and recommend ICD-9 codes based on them [11]. MedLEE is another NLP-based data collection system that has also been operational in a hospital setting which has been shown to be accurate in extracting age, gender, race, and ethnicity when its search scope was limited to specific sections [12]. MedLEE has also been extended to detect medical problems in discharge summaries [13]. Results have not shown these applications' abilities to specifically extract marital status and education level however, and the literature is not clear on these applications' use in psychiatry-based documents.

Prior research in extracting marital status and education level is also available but limited. Marital status extraction was included as part of a study involving a corpus of German curriculum vitae with help from Hidden Markov models; performance was very strong [14]. Another study examined the Social History section of clinical notes and found that marital status was one of the more common types of information found, showing that the Social History section is a preferable section to search for such information. The study also examined the ability of several types of classifications including HL7 and openEHR to code marital status and the extent to which the classifications aligned with each other [15]. For extracting education level, one study was found which used Naïve Bayes, Support Vector Machine, and K-Nearest Neighbor classifiers to extract education level from Facebook pages; the best result came from the Naïve Bayes classifier with an accuracy of 86.15% [16]. Automatic extraction approaches have not yet been explored for marital status and education level using clinical texts in either the medical or psychiatric field however, where patients' social histories may be more complex when expressing these kinds of information.

Although some of these demographic extraction tasks used machine learning, these studies also utilized much larger corpuses than the one made available for this current study. A larger corpus allows for a greater variety of patterns to be detected using a supervised machine learning approach as well as more effectively evaluate generalizability, although a system could be tuned with a smaller corpus using partitioning and cross validation if necessary. These studies also did not discuss specific features that were considered in training the machine learning algorithms.

The focus of many of these studies was also not purely on extracting demographics and the results of demographics extraction were not reported and discussed in as much detail as it will be during the present study. Due to the lack of in-depth research in extraction of a variety of types of demographics from a corpus of structured and unstructured text, it was decided that a rule-based approach would be developed first in order to establish a baseline for future studies using machine-based approaches. In addition to using rules based on keywords, rules utilizing negation detection and dependency parsing will also be used in our approach when extracting information from free-text.

2.4 Background in Rule-based Methods in Clinical Text Mining

Rule-based approaches have been used in clinical text mining tasks in the past as methods for negation detection. NegEx is a simple rule-based algorithm that has been used for negation detection and is based on regular expressions [17]. It has been used in studies to identify obesity and its comorbidities [18], identify risk factors for sudden cardiac death using ECG data [19], and has been tested on pathology reports [20]. When tested on a corpus of sentences from clinical discharge summaries, it achieved a sensitivity of 94.5% and a specificity of 77.8%. However, it only detected negation in concepts that have first been mapped to UMLS [17].

Another rule-based negation detection algorithm is Negfinder, which also uses regular expressions to identify negation signals. It also uses grammar rules including a single token Look Ahead Left-Recursive grammar to determine whether the negation signal applies to one or more concepts and if the concepts precede or succeed the negation signal. The grammar rules are based on the relationship between the identified concept, the negation signal, negation terminators, sentence terminators, and any other terms considered to be “filler”. It does not perform any deeper parsing of the sentence structure and has a reported sensitivity of 95.3% and specificity of 97.7% [21]. Huang and Lowe also developed a method using regular expressions to match for negation signals in conjunction with grammatical parsing, but they used a more sophisticated parsing method in comparison to Negfinder by using syntactic patterns found in parse trees. Their approach achieved a sensitivity of 92.6% and a specificity of 99.87%, indicating that the addition of deeper syntactic parsing can play a part in achieving greater precision in negation detection [22].

Rule-based negation detection approaches have also been enhanced through the inclusion of dependency parsing to utilize patterns found its deeper syntactic and semantic representations. Dependencies are binary relations between words in a sentence, where one word acts as the "head" and another word takes the role of a "dependent" that relies on the head for modification or specification. Unlike phrase structures based on constituencies, dependencies do not feature nodes indicating phrasal categories and are only based on lexical categories instead [23].

Sohn, Wu, and Chute explored the use of dependency paths in negation detection by using a modified version of the dependency parser found in cTAKES and was known as DepNeg. The dependency path patterns were based on the syntactic relationship between the concept and the identified negation signal words. When DepNeg was compared to the original negation module found in cTAKES, which itself was based on NegEx, it was found to be superior in F-score (DepNeg: 0.838 vs. cTAKES: 0.822) and accuracy (DepNeg: 0.946 vs. cTAKES: 0.934) [24]. The present study aims to utilize dependency parsing when negation detection is needed to identify and properly code demographics.

Dependencies can be presented as either projective or non-projective. Projective dependencies present each word in a sentence besides the head as a dependent of another word. When presented as a graph with lines tracing the head of each word to its dependent, projective parses will not have any crossing or overlaps between the lines. Non-projective dependencies allow for overlaps in such graphs, which allows more flexibility in expressing direct word relations. This becomes especially beneficial for relations involving the referent of a relative clause or dependencies involving prepositions, and can be useful in languages with flexible word orders. Because the output of a non-projective dependency parser condenses several dependencies into a single dependency, this approach is also known as a “collapsed” representation [25, 26].

The use of collapsed dependencies has been found for clinical text mining in the extraction of diagnoses of family members in clinical assessments. Because identifying the correct person that the diagnosis is referring to is important in this task in addition to the diagnosis, Lewis, Gruhl, and Yang utilized dependency parsing using the Stanford Parser [27] in order to extract a diagnosis mapped to an ICD-9 code as well as the family member it relates to. The algorithm was based on a series of rules using collapsed dependencies in order to take advantage of patterns in semantic context. The algorithm achieved a reasonably high recall and precision during development, although recall was not as high using the test set due to the reliance on Named Entity Recognition (NER) to identify diseases [28].

Lewis, Gruhl and Yang extended this task by analyzing the patterns found in phrase structures and typed dependencies to determine the presence of a disease in the patient’s family history. When compared to the 49% of the targeted family history that was identified in the training set using the top ten rules based on only the phrase structure, the top ten dependency paths were able to identify 57% of the targeted family history. By utilizing dependency structures rather than NER, they were able to increase recall and consequently precision in comparison to their previous study [29].

Outside of extracting diagnoses, dependency parsing has also been notably used in extracting biological events. RelEx is a rule-based approach that utilizes dependency parsing to extract relations between genes and proteins and has a reported precision and recall of 80% in extracting gene-protein relations [30]. It also has been used to extract protein-protein interactions [31]. A different approach utilizing the Stanford Parser to extract biological events has also been explored and resulted in a reasonably high overall precision but low recall [32]. Compared to the extremely wide variety of concepts potentially involved in biological events however, demographics are expected to be expressed with much less variability. This should make them more suited for dependency parsing using the Stanford Parser.

2.5 Conclusion

This study has the opportunity to fill a gap currently present in the research literature regarding both text mining of demographics as well as text mining approaches using clinical texts in the psychiatry domain, especially in regards to identifying diagnoses. This study will determine whether a small set of simple rules can be used to correctly extract and code psychiatric diagnoses made by a clinician in addition to demographics such as age, gender, education level, and marital status. The use of a rule-based approach will also determine if similar kinds of rules can be used to extract different kinds of demographics. The overall goal of this study is to develop a text mining approach using psychiatric assessments that could potentially be used in a clinical research setting.

Chapter 3: General Methods

This chapter will discuss the development and evaluation of the annotation schema used in this study. It will also describe the general-purpose section parser and automatic evaluator used for all of the text mining algorithms used in this study. Lastly, the metrics used for evaluation throughout the study are explained.

3.1 Development of the Gold Standard

When developing a text mining approach, it is important to be able to evaluate it. One evaluation method is to create an annotated version of the corpus to use as a gold standard. This gold standard can be used to determine if all of the targeted features had been extracted. Developing a gold standard first requires collecting the corpus that will be used for the extraction tasks and then developing a schema to ensure consistent annotation across the corpus. The corpus used in this study consisted of 200 admission psychiatric assessments collected from an eating disorders unit at a mental health hospital. The assessments were from 200 consecutive adult patients admitted to the unit between 2011 and 2012 for whom admission psychiatric assessments were available in their electronic medical record. All protected health information was manually de-identified in each document by replacing them with a generic placeholder.

3.1.1 Description of the Corpus

Each category was consistently found in a particular section of each document throughout the corpus, which is described in further detail below:

- **Gender** was found in a structured format in the header of the document, starting with the prefix “SEX:” and the followed by either an “M” for male or an “F” for female.
- **Age** was found in the narrative Chief Complaint, History of Present Illness, Identifying Information, or Impression sections. An example is shown in bold below:

  HISTORY OF PRESENT ILLNESS: The patient is a **19-year-old** who has been in treatment for anorexia nervosa, OCD, and depression in the past. The patient reports never feeling comfortable in her body and would typically worry about food, weight and shape. She has a diagnosis of major depressive disorder made her sophomore year of high school. She has been treated in the past with Lexapro and Abilify. No history of alcohol or drug abuse.
- **Admission Diagnoses** were always found in a table at the end of the assessment amongst the Axes I through V diagnoses which are typically used in psychiatric treatment. Our focus is only on the Axis I diagnoses that describe the patient’s specific psychological disorders. An example of one of these tables as it is found in the corpus is shown below in Table 1:

Axis I:	(1) Bulimia nervosa. (2) Major depressive disorder, recurrent, current episode moderate to severe with passive suicidal ideation. (3) ADHD, by history.
Axis II:	Deferred.
Axis III:	None acute.
Axis IV:	Moderate to severe.
Axis V:	Current Global Assessment of Functioning - .45. Highest in the past year – 70 to 80.

Table 1: Example of a Typical Diagnosis Table

- **Marital Status** and **Education Level** are both found in the narrative Social History section in an unstructured format, making them the most challenging to extract and code. Examples of the patient's current marital status and education level are shown in bold below:

> SOCIAL HISTORY: She was born and raised in the <PHI>LOCATION</PHI> area, currently living in <PHI>LOCATION</PHI>. She did finish high school at a therapeutic boarding school. **She is currently a freshman in college and looking to study psychology**. **Has few friends and has difficulty with developing intimate relationships, and as a result is currently single**. She denies any history of physical or sexual abuse.

3.1.2 Annotation Schema Development

Establishing a comprehensive annotation schema is important in order to ensure consistent annotation across all the documents in the corpus so that a gold-standard can be developed for evaluation. Development of the annotation schema had two goals based on the objectives of the study:

1. Define the correct codes that should be assigned to each document.
2. Determine the spans of text that could be used to determine each code.

Coding Guidelines

The codes for this project were based on the coding guidelines already being used at the hospital involved in this study. It was developed through an extensive review of patient charts and has evolved alongside the clinical research needs of the hospital. Because these guidelines have already been developed over a lengthy period of time, it can be assumed that they comprehensively apply to the majority of potential patients and were considered sufficient for use in this project. The assessments sometimes did not provide enough conclusive information to determine a particular code, and in those cases that category was left un-coded in that assessment. In order to better ensure inter-annotator agreement in classifying assessments with the correct codes, it was included in the schema guidelines that the annotator was to only determine the codes based on the information provided in the text span selected by the annotator. That is, the annotator was not to make any assumptions beyond what is presented in the assessment or make any judgments about the reliability of the information provided by the patient in the assessment.

Text Span Selection Guidelines

The text span selection component of the schema was developed in order to determine whether the algorithm inferred the code using the correct span of text. Because gender was always present in the structured portion of the assessment, text selection was straightforward. Diagnosis was also provided in a structured format in the assessment as seen in Table 1. Diagnoses that were inconclusive, in remission, or ruled out were excluded. For example, the text “ADHD, by history” in Table 1 would not be selected or coded. Any specifiers associated with annotated diagnoses were also to be selected to simplify annotation. In the example of Table 1, the diagnosis “Major depressive disorder” would be selected in addition to its specifier “recurrent, current episode moderate to severe with passive suicidal ideation.”

As described earlier in this chapter, age, marital status and education level were provided in the unstructured free-text portion of the assessment, and for these categories a rule was established that only the most minimal amount of text should be selected that would still sufficiently provide enough information to determine the coding. The schema includes text span guidelines for three separate cases depending on the structure and complexity of the sentence:

1. If possible, the beginning of the text span was to always start with the noun phrase that refers to the patient.
2. If the noun phrase was not available but the patient was clearly implied as a subject, the beginning of the first verb phrase was used.
3. Occasionally there were sentences without a subject or a verb provided that still were helpful, such as “Currently married.”, and in those cases the first relevant word was the start of the text span.

In regards to the examples found in Figures 1 and 2, the text span selected for age in Figure 1 would be “The patient is a 19-year-old**”.** In Figure 2, the text span selected for marital status would be “Has few friends and has difficulty with developing intimate relationships, and as a result is currently single” and the text span selected for education level would be “She is currently a freshman in college”.

The end of the text span was dependent on the annotator’s judgment with respect to the rule that the minimal amount of text was to be selected while still providing enough information to determine the correct code. Periods at the end of the sentence were not annotated in case the sentence splitter involved in the algorithm removed the period while splitting. Since the primary goal of this project is to determine the correct coding, the selected text span is only necessary for evaluating whether the text extracted by the algorithm matches the annotated text in the gold standard. As a result, only a partial text match was needed as long as the complete annotated text span was found in the extracted text. This allows for some flexibility in the amount of text selected.

Although the text mining algorithm had not been fully developed at the time that the schema was developed, the schema was based on some general assumptions about what the algorithm's overall architecture would be like. Namely, that the algorithm would use a section parser and a sentence splitter. This would result in each section as well as each individual sentence within each section being processed independently of each other. Because of this a decision was made to annotate all occurrences of a particular category in the free-text section(s) that the algorithm would check. This was to ensure that the evaluation process was based on the behavior and output of the algorithm.

Annotation Evaluation

In order to determine the quality of the annotation, a subset of the corpus was annotated by two reviewers as a double-blind test. The primary annotator (PA) was the author of this study and the secondary annotator (SA) was a female employee of the psychiatric hospital from which the corpus came from. Both the primary and secondary annotators had previous experience in reviewing psychiatric assessments and were familiar with reviewing psychiatric diagnoses and patient demographics. The annotation completed by the PA was considered to be the gold standard. The PA trained the SA by using set of guidelines in order to more effectively communicate the details of the annotation schema.[1] All annotation was completed using eHOST, an open source annotation tool.[2]

After the schema was developed, the co-annotator agreement was determined through two iterations of evaluation. The first iteration involved comparing the SA's annotations on 10 assessments to the PA's gold-standard annotation. The second iteration involved the same comparison but with another 20 documents after revisions to the annotation guidelines. Determining co-annotator agreement involved calculating Cohen's kappa for coding agreement and calculating partial match accuracy for the selected text spans. Accuracy was calculated using the following formula:

$$accuracy = \frac{true\ positives + true\ negatives}{true\ positives + false\ positives + false\ negatives + true\ negatives}$$

A partial overlap in text spans between the PA's gold standard and the SA's annotation was considered to be a true positive, a text span selected by the SA that did not overlap the PA's gold standard at all was considered a false positive, and a text span that should have been selected but was completely missed by the SA was considered a false negative.

The first round of co-annotator evaluation using 10 documents yielded satisfactory coding agreement in all data categories (Table 2), with a kappa value between 0.6 and 0.8 considered as acceptable. Because all 10 assessments were

[1] The annotation schema guidelines are found in the Appendix.

[2] https://code.google.com/p/ehost/

regarding female patients, the gender category was constant and kappa was not calculated. Kappa was also not calculated for quinary diagnosis because none of the patients had more than four coded diagnoses. One error was made by the SA in the tertiary diagnosis category, where in one assessment the code for “Bipolar Type 1” was assigned when it should have been “Bipolar NOS”. Two errors were made in the marital status category; both involved assessments that were coded “Unspecified” when the assessment had indicated that the patient either did not currently have a spouse or that they had a boyfriend, both of which are sufficient to classify the patient as “Single”. None of these errors warranted a revision of the annotation guidelines, but the SA was made aware of them.

For text span selection, partial match accuracy was generally acceptable (Table 3). An error made in the age category was due to an age being selected in the wrong section that would not be targeted by the text mining algorithm. In the education level category, one error was made due to a selected phrase that mentioned that the patient received a degree but not whether it was a Bachelor’s degree or from a higher level of graduate schooling. Only marital status presented a low text span selection agreement; two errors were false negatives that the SA did not consider to be indicative of marital status, and one error was due to a selected phase that indicated a patient’s past marriage although the patient was currently divorced. Since the algorithm would be evaluating sentences independently of each other, the sentence indicating that the patient was married may result in an incorrect coding, which is why it was not selected in the gold standard. Only indications of the patient's current marital status was to be selected based on the annotation guidelines.

Although accuracy for marital status was much lower than the other categories, all of the errors were preventable by referring to the annotation guidelines. After evaluation the SA was made aware of these errors and pointed to where clarification could be found in the annotation guidelines. No changes were made to the annotation guidelines as a result of this first round of iteration, and the prediction was that performance would further improve in future evaluations due to a practice effect.

Because the annotation schema was considered sufficient based on the results of the first iteration of evaluation, a final co-annotator comparison of 20 more assessments was completed. The results indicated sufficient co-annotator agreement in both coding and text span selection (Table 2, Table 3). Again none of the assessments had more than four coded diagnoses so kappa for quinary diagnosis was not calculated. Kappa increased or stayed the same in all variables except for quaternary diagnosis and education level. A quaternary diagnosis of "Anxiety, Not Otherwise Specified with Trichotillomania" received codes for both "Anxiety NOS" and "Trichotillomania" when Trichotillomania is considered only a diagnosis specifier in this case. The additional diagnosis code in this case also resulted in a lower kappa for tertiary diagnosis. The SA also continued to use the "Bipolar Type 1" coding rather than "Bipolar NOS" in an instance where no subtype was stated. There also was one error in coding education level, where the level of education was unclear but the co-annotator coded the patient as "High School Graduate" based on the assumption that the patient was not currently in school. There also was one error in coding a patient's marital status as "Single," where the co-annotator made an assumption based on the patient's reported lack of social skills and difficulty with intimacy rather than a specific mention of the presence of a significant other.

Partial text span accuracy also remained the same or improved. There was one missed instance of age on one assessment but accuracy was still strong. There was one false positive selection for education level as well as marital status, which are related to the previously mentioned coding errors.

	Gender	Age	Primary Dx	Secondary Dx	Tertiary Dx	Quaternary Dx	Quinary Dx	Education Level	Marital Status
1st Round (10 Documents)	N/A	1	1	1	.855	1	N/A	1	.583
2nd Round (20 Documents)	1	1	1	1	.87	.783	N/A	.928	.924

Table 2: Cohen's Kappa Values of the Double-blind Co-annotator Study

	Gender	Age	Primary Dx	Education Level	Marital Status
1st Round (10 Documents)	1	.95	1	.9	.4
2nd Round (20 Documents)	1	.95	1	.95	.92

Table 3: Co-annotator Partial Match Text Span Accuracy

Overall, these results indicate that inter-annotator agreement is high enough to support the validity of the gold standard. It is remarkable that the final results were so strong considering the subjectivity of some categories, specifically marital status and education level, which are only found in the free-text sections and requires some inference. The goal of the annotation schema was to limit this inference in such a way that different annotators would provide the same results, which was accomplished by restricting the annotated text to certain sections as well as by only annotating text directly related to the patient's current status. This was in addition to attempting to cover the wide variety of ways that such information can be semantically expressed in a narrative format. The coding guidelines were similarly designed with generalizability in mind while also aiming to account for observed exceptions; the annotation schema sufficiently met this goal as well. Because the annotation schema was considered to be suitable, the rest of the corpus was annotated based on this schema.

3.2 General-Purpose Algorithms

Two general-purpose algorithms were developed which were used in the extraction algorithms for all of the categories: a *section parser* and an *automatic evaluator*. The section parser is a simple but crucial algorithm developed based on patterns found in the corpus that indicate individual sections. The section parser is used to identify and extract specific segments of a document which are then processed by the text mining algorithms. Restricting the focus of a particular text mining algorithm increases the precision and efficiency of the algorithm, although it is based on the expectation that a particular piece of information will always be found in a targeted section.

The second algorithm developed to assist with the study is an automatic evaluator, which compares the text mining algorithm's output to the annotated gold standard to determine the algorithm's accuracy metrics. Each algorithm's output for each document contains the determined code, the final source text segment used to ascertain the code, and a list of all the possible source text segments identified as being related to the targeted type of information. The XML file from eHOST, which was used to annotate the documents, was parsed using the *etree* module in the lxml XML toolkit in Python[3] to extract the annotated text and codes from the gold standard. The source text segments and codes were compared using partial string matching. The results of the comparison to the gold standard results in the calculation of several metrics: code accuracy, text accuracy, recall, precision, and F-score which are all explained in Section 3.3.

The automatic evaluator was found to be very accurate and reliable during the study, with only marginal errors identified as being due to errors caused in reformatting the text file or from processing by the sentence splitter.[4] It was considered appropriate for use in evaluating all of the algorithms besides gender, which required manual evaluation.[5]

3.3 Evaluation Metrics

There are several metrics that were calculated to determine the accuracy of each algorithm. The code accuracy indicates the percentage of documents that the algorithm had assigned the correct code to and is calculated as follows:

$$Code\ accuracy = \frac{Number\ of\ correct\ documents\ coded}{Number\ of\ total\ documents\ evaluated}$$

[3] http://lxml.de/

[4] Further details are provided in the discussion of “Limitations” in Chapter 7.

[5] Further details are provided in the discussion of “Gender Extraction” in Chapter 6.

Text accuracy shows the extent to which the *final* source text segment chosen for each document to determine the code is correct based on the gold standard:

$$Text\ accuracy = \frac{Number\ of\ documents\ coded\ with\ positively\ matched\ text}{Number\ of\ total\ documents\ evaluated}$$

Recall shows the extent to which the initial pattern matching stage of the algorithm is extracting *all* of the possible text segments from which the correct code may be chosen. True positives were considered to be source text segments that were identified by the text mining algorithm which were also found in the gold standard. False negatives were source text segments found in the gold standard which were not identified by the text mining algorithm:

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

Precision shows the extent to which *only* the relevant sentences are positive matches. False positives were considered to be source text segments that were identified by the text mining algorithm which were not found in the gold standard:

$$Precision = \frac{true\ positives}{true\ positives + false\ positives}$$

The F-score is the harmonic mean of recall and precision:

$$F - score = 2 \cdot \frac{precision\ \cdot\ recall}{precision\ + recall}$$

These metrics are used in evaluations of all of our methods.

Chapter 4: Education Level Extraction and Coding

4.1 Introduction

This chapter details the development and evaluation of a rule-based algorithm to automatically detect text that expresses the patient's most recent education level, and then, classify the document with a code based on the identified text. Section 4.2 discusses the development of the algorithm which occurred in two phases: a keyword-only phase and a phase utilizing natural language processing (NLP). Each phase also includes error analyses of the development set. The results of each phase over the test set are presented and discussed in Section 4.3. The goal of this part of the study was to determine whether using keyword-based rules would provide a sufficient baseline for this extraction task, and then to explore if rules based on NLP were necessary and would show an observable improvement in performance.

Education level is a type of demographic information that is expressed in the narrative/free-text section of the psychiatric assessment. The hospital at which this study took place collects this information as part of its outcome studies research. This information could be used to examine the relationship between mental disorder symptoms and education level and how it impacts quality of life. Education level is currently manually abstracted from the psychiatric assessment and assigned the closest relevant code, which is stored in a database. The possible classifications are as follows:

Some High School – This code is used when the psychiatric assessment indicates that the patient has enrolled in high school (or has enrolled in a high school equivalency program such as to obtain a GED) but has not graduated or completed the program at the time of hospital admission.

High School – This code indicates that a patient has graduated from high school or received their GED. This code is also used when the psychiatric assessment indicates that the patient has enrolled in college but does not further clarify that the patient has begun classes in the college.

Some College – This code is used to classify patients whose psychiatric assessment indicates that they are currently taking college classes but they have not yet graduated or it is not clear that the patient has graduated. For example, if the assessment merely states that the patient attended college with no explicit indication about graduation, the education level is coded as “Some College”. This code is also used for patients who have taken a leave of absence or withdrawn from college.

Associate’s Degree Graduate – This code is used to classify patients whose psychiatric assessment indicates that they have graduated with an associate’s or two-year degree. Patients who are working towards such a degree but have not graduated or it is not clear that they have graduated are coded as “Some College”.

College (Bachelor’s) Graduate – This code is used to classify patients who have graduated with a Bachelor’s degree. Patients who are working towards such a degree but have not graduated or it is not clear that they have graduated are coded as “Some College”.

Some Graduate School – Similar to the “Some College” classification, this code is used for patients whose psychiatric assessment indicates that they are currently working towards some sort of post-undergraduate degree but they have not yet graduated or it is not clear that the patient has graduated. This applies to both the Master’s and Doctoral levels.

Graduate School (MS, MA Level) – This code specifically indicates patients who have received a Masters’ degree.

Graduate School (PhD, JD, MD, etc. Level) – This code specifically indicates patients who have received any degree beyond the level of a Masters’ degree. This typically is some sort of Doctorate degree which includes the following: PhD, JD, MD, Pharm. D, and PsyD.

Unspecified - This is used to indicate that the patient’s education level could not be clearly determined from the psychiatric assessment.

4.2 Method

4.2.1 Phase 1 – Keyword Based Algorithm

The main goal of this phase was to maximize recall of the keyword matching stage of the algorithm as much as possible. The secondary goal of this phase was to increase coding accuracy once a sentence was matched for a certain education level through the use of rules based on additional keyword matching. A particular rule was not implemented to increase coding accuracy if it would be at the expense of recall however. The algorithm was developed through an iterative process where a portion of the development set was used to develop the algorithm while a separate set of 20 documents from the development set were set aside as a dev-test set (Table 5).

Once the algorithm was run on the dev-test set, it then became a part of the development set for the next iteration as errors were analyzed and used to further develop the algorithm. The revised algorithm was then evaluated with another unused dev-test set of 20 documents, which were also added to the development set for the next iteration and so on. This approach allowed much of the corpus to be used for both development and evaluation and was adopted due to the relatively small size of the corpus. This process also allowed the recall of each iteration to be compared against each other to determine how many documents were needed before recall would not increase any further. There were four development iterations and one final evaluation iteration with a test set of 90 documents that had not yet been used during development.

The iterative development process was used to refine the list of keywords that would correctly identify sentences that were relevant to current education level. Reoccurring errors that could not be resolved by using keyword-based rules alone were also identified during this time. These errors would be targeted with a NLP-based approach in the second phase of development. The rules for coding the document were also developed during this process as well.

A flow chart describing the general sections of the algorithm can be found in Figure 1. The algorithm begins by processing the documents with a sentence splitter and then a section parser in order to extract the sentences in the Social History section of the assessment. The scope of the algorithm's search was restricted to just the Social History section because it was observed that education level was most often expressed in this section. Although the patient's educational history occasionally appeared in the section discussing the patient's clinical history, the most current education level was usually not as clear as what was found in the Social History.

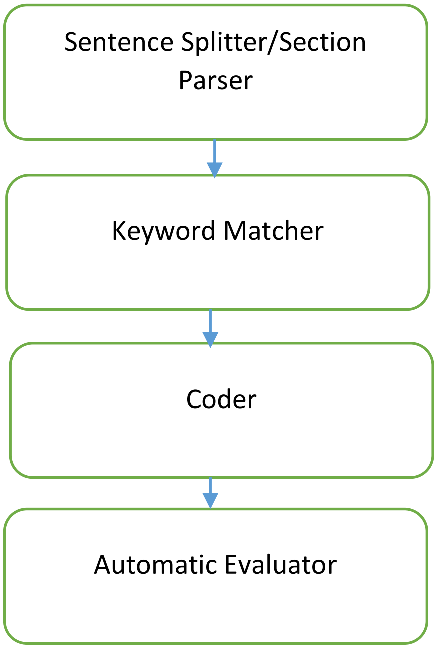

Figure 1. High-Level Abstraction of the Keyword-Based Education Level Extraction Algorithm.

Extra white space and newline characters were then stripped from each sentence and the first character in the sentence was made lowercase. Since using case-sensitive regular expressions was the primary method of keyword matching used in Python, this reduced the chances of a potential false negative due to capitalization (e.g. "Graduated from college" vs. "graduated from college"). Each sentence was then individually searched with a series of regular expressions based on keywords that indicate code groupings of certain levels of education in the following order: High School (Some or Completed), College (Some, Associate or Bachelor's degree), and lastly Graduate School (Some, Graduate School MA/MS Level, Graduate School PhD/MD/JD/etc. Level). If a document did not contain any sentences that matched at any of the education levels, it was automatically coded as "Unspecified".

The regular expressions used at this level in the final evaluation stage are in Table 4. They are based on observation of the development set of 110 assessments over the 4 iterations, and our own ideas about possible expressions used to express education level. To expand the regular expressions further, we utilized synonyms from WordNet 3.1.[6]

High School	high school, HIGH SCHOOL, [^\w](\s*)GED
College	TECH, technical, associate, two-year, college, COLLEGE, university, community college, major, four-year, (B|b)achelor('*)s, economist, ology, degree, kicked out, dropped out, withdrew, leave of absence
Graduate School (MA/MS Level)	graduate student, graduate school, GRADUATE, [^\w](\s*)MBA, [^\w](\s*)MA, [^\w](\s*)MS, (M|m)aster('*)s, postgraduate
Graduate School (PhD/MD/JD etc. Level)	[^\w](\s*)PhD, [^\w](\s*)MD, [^\w](\s*)Pharm. D, [^\w](\s*)PsyD, dissertation, medical school, law school, postgraduate

Table 4: Regular Expressions Used to Extract Education Level

[6] http://wordnet.princeton.edu/

The code groupings were ranked in such a way that the keywords matched at the highest-ranked code grouping were processed further for coding. The rank from lowest to highest was as follows: High School, College, Graduate School (MA/MS Level), and Graduate School (PhD/MD/JD etc. Level). For example, "The patient graduated from high school and is currently in college" would match at both the High School and College keyword levels, but because the College keywords are ranked higher, only the codes associated with College will be considered. The precision, recall and F-score metrics were also calculated at this point.

If a sentence was matched for a particular education level, it is then processed with a series of pattern matching rules to determine the correct code for that education level. These rules are based on matching for additional regular expressions that evolved over the course of algorithm development. If the sentence satisfied a particular rule, it was given the associated code if it was in a higher-ranked code grouping than the code currently assigned (all documents begin coded as "Unspecified" which has the lowest ranking). This is the only way that other sentences in the document had an effect on how the current sentence is processed. For example, if a previous sentence resulted in a coding of "Some Graduate School" and the current sentence resulted in a coding of "College Graduate", the coding based on the current sentence was ignored. After a code has been assigned, the last sentence used to determine the code is compared to the set of annotated text segments in the gold standard. This is to determine the text accuracy of a certain set of documents during evaluation.

The performance of the algorithm over the dev-test set at each iteration is presented in Table 5. As discussed in Section 3.3, recall, precision, and F-score reflect the performance of the keyword-matching stage before coding occurs. Code and text accuracy reflect the algorithm's ability to determine the correct code from the correct text. Recall increased with each iteration, indicating that the algorithm continued to benefit from keywords found in additional documents. The benefit of a larger development set is also evident in the code and text accuracy, which both fluctuated until settling at 0.90 in the fourth iteration. The fluctuation is due to adjustments made to the coding rules which continued throughout development.

Iteration	Development Set Size	Dev-Test Set Size	Code Accuracy	Text Accuracy	Recall	Precision	F-Score
1	30	20	0.60	0.80	0.73	0.64	0.68
2	50	20	0.55	0.85	0.89	0.70	0.78
3	70	20	0.75	0.80	0.91	0.63	0.74
4	90	20	0.90	0.90	0.94	0.79	0.86

Table 5: Dev-Test Set Performance of the Keyword-Based Education Level Algorithm

Error Analysis

Qualitatively, there were several common types of errors found when analyzing the errors from the dev-test set at each iteration, many of which were not easily resolved by using keywords alone. This is why some error types appear several times throughout the iterations. A summary of the types of errors found in each iteration can be found in Table 6. The error types are also described in further detail below.

	Dev-Test Set Errors
1st Iteration	1, 2, 3, 5, 6, 7, 8
2nd Iteration	1, 7, 8
3rd Iteration	1, 2, 4, 8
4th Iteration	2, 4, 7
Error code key: 1 = Multiple education levels; 2 = Non-patient subject; 3 = Negation; 4 = Modality/speculation; 5 = Missing keywords; 6 = Sentence splitter error; 7 = Sentence too vague; 8 = Syntactic relation error	

Table 6: Dev-Test Set Error Summary of Keyword-Based Education Level Algorithm

Error 1: Multiple Education Levels

The most common type of error found were sentences that contained keywords that matched for multiple levels of education, which also was the cause of some syntactic relation errors. For example:

> "The patient does have one sibling and currently has been a student at ABC-**COLLEGE** and **graduated** from **high school**…"

On the basis of this sentence, the keyword-based algorithm would determine the code to be "College Graduate" due to the presence of the keywords "COLLEGE" and "graduated." However, "graduated" is more closely related to high school rather than college, indicating that the code should rather be "Some College" since the patient is currently still attending classes at college. This type of error requires deeper syntactic processing using NLP.

Error 2: Non-Patient Subjects

Another common error was due to sentences with keywords that indicated a certain education level but were predicating about a member of the patient's family rather than the patient themselves. For example the following:

"She has a **brother** who is a freshman in college."

This sentence indicates that someone is currently attending college but the phrase is predicating about the patient's brother rather than the patient. This type of error will also be addressed in the next phase through the use of NLP.

Error 3: Lack of Negation Detection

There were also issues related to the algorithm's inability to properly detect negations. For example,

"The patient….did **not graduate** from **high school**."

The algorithm would find a positive match for "high school" as well as "graduate", which resulted in a code of "High School Graduate." Since the verb phrase "did not graduate" contains a negation relation between "not" and "graduate" however, it requires further processing to determine a more accurate code.

Error 4: Modality/Speculation

There were some sentences that also matched for certain education level keywords but they were related to the patient's possible future intention to attend school. These sentences are only speculative however and should not be considered as a positive match. They often contain a verb phrase with a modal verb such as "would like to." For example:

> "She would like to go on to **graduate school**."

Although this sentence matches for "graduate school," it is not clear on whether the patient had actually started taking graduate classes or not and should not be considered eligible for coding.

Additional Errors

Other errors that occurred throughout the iterations during development included sentences that were false negatives due to the necessary keyword being missing from the algorithm, which was resolved in the next iteration. There was one error that could be attributed to the sentence parser, which misinterpreted the punctuation mark in the degree title "Pharm. D" as a period denoting the end of a sentence, which then prevented the algorithm from finding a positive match for that particular degree title due to the "D" being delegated to another text line. Lastly, some sentences were annotated in the gold standard as positive matches but were considered to be too vague to be indicative of a clear education level on their own. These sentences often did not have any discernable keywords and usually provided context to other sentences indicating education level in the assessment. Since these kinds of errors were not considered resolvable in the algorithm, they were set aside.

4.2.2 Phase 2 – Addition of NLP-Based Rules

While the primary goal of the first phase was to increase recall, the goal of this phase was to use syntactic and semantic processing to develop NLP-based rules that will increase the precision of the algorithm as well as increase coding and text accuracy. NLP rules were informed by error analysis of the results of the keyword-based algorithm on the development set.[7] These rules were designed to examine the syntactic context of certain keywords in order to interpret keywords matched in the sentence.

The Stanford Parser [27] was used to generate typed dependencies and constituent parses from the text strings in order to utilize deeper semantic and syntactic processing for these new rules. In order to incorporate these additional processes into the algorithm, some restructuring was required. A high-level flow chart of the restructured algorithm can be found in Figure 2. Pre-processing is the same as the previous phase in regards to utilizing the sentence splitter and section parser as well as preparing the text for keyword-matching.[8] It is after this pre-processing but before the keyword matching stage which is when the majority of the NLP-based processing occurs.

[7] Further details are provided in "Error Analysis" in Section 4.2.1.
[8] Further details are provided Section 4.2.1.

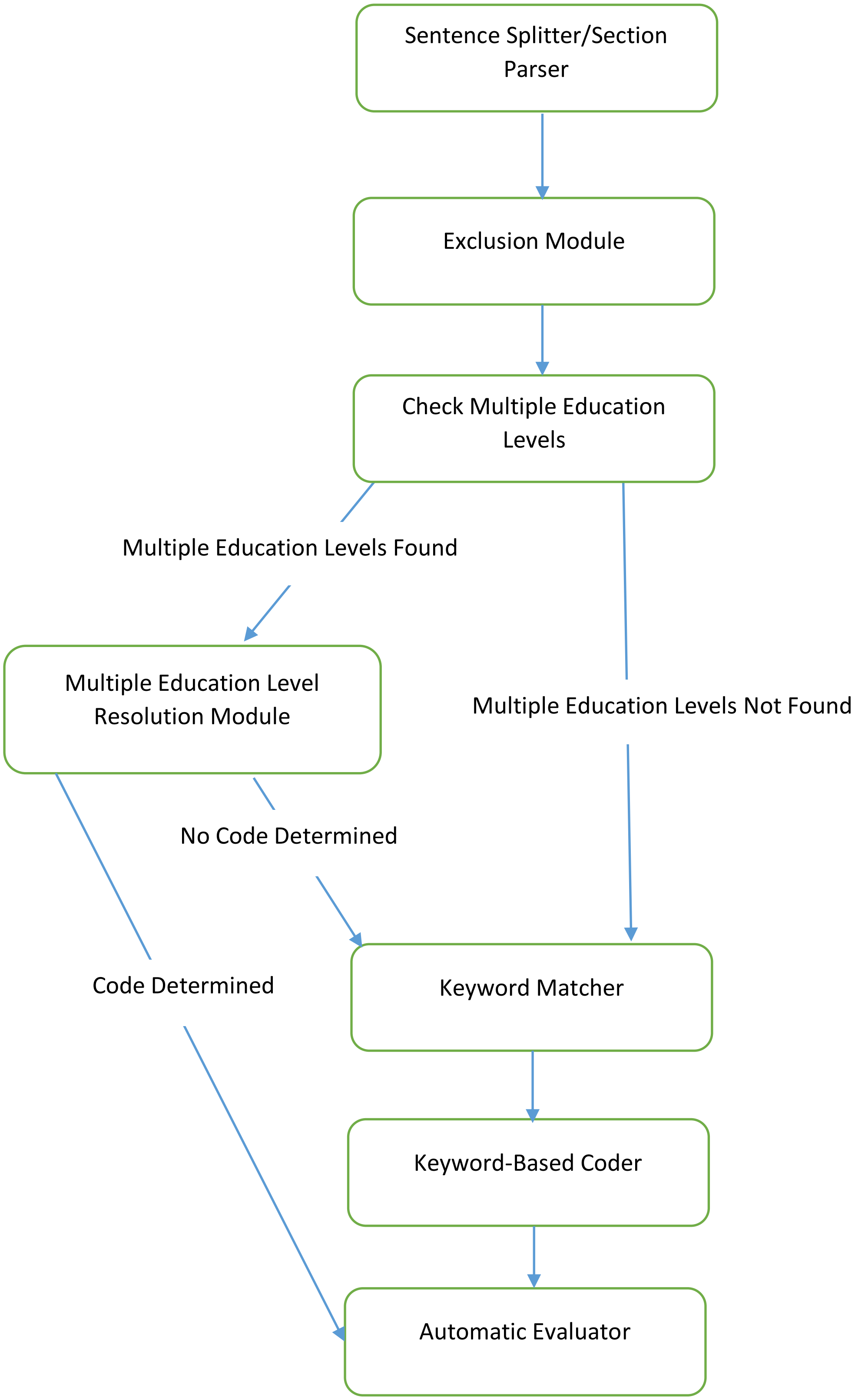
Sentence Splitter/Section Parser
Exclusion Module
Check Multiple Education Levels
Multiple Education Levels Found
Multiple Education Level Resolution Module
Multiple Education Levels Not Found
No Code Determined
Code Determined
Keyword Matcher
Keyword-Based Coder
Automatic Evaluator

Figure 2. High-Level Abstraction of the NLP-Based Education Level Extraction Algorithm.

Exclusion Module

The first new feature utilized in the workflow was an exclusion module. This was developed due to two error patterns that were observed during development of the keyword-only approach in the first phase. The first error pattern was based on sentences which were identified by the keyword algorithm as containing terms related to education level, but were not referring to the patient. A second error pattern involved statements of speculation regarding a patient's intention to attend a particular education level in the future. This exclusion module aimed to resolve both of these error patterns by identifying sentences that match either of these error patterns and then either completely excluding them from the keyword-matching stage (and subsequently coding) or excising parts of the sentence that may cause errors at the keyword-matching stage.

In order to correctly resolve sentences that featured someone other than the patient as a subject, a keyword list of nouns commonly describing potential subjects that were not the patient such as parents, siblings, and other family members was first developed. All sentences were searched for these keywords; if none of the keywords were found it was not processed at all by the exclusion module and was passed onto the next stage. For sentences that positively matched for one of these keywords, two separate sets of exclusion rules were developed for simple single clause sentences and complex multiple clause sentences.

Exclusion Module – Non-Patient Individuals in Single Clause Sentences

For single clause sentences, dependency parses were used to determine who the education level was being predicated about. Rules based on observed patterns in the dependency parses were developed and tested, and it was determined that the most effective rules targeted either a nominal subject or a direct object of the sentence. This list was used to search through the "nsubj" (nominal subject) and "nsubjpass" (passive nominal subject) nodes of the dependency parse for the root of a clause or the "dobj" (direct object) node for the accusative object of a verb.[9]

The nominal subject was considered because many sentences did not refer to the patient at all. For example,

"**Father** was a **college** professor." → nsubj(professor-5, Father-1)

This sentence matches for they keyword "college", however the only subject in the sentence is the word "father," which is one of the keywords indicating a non-patient subject. As a result, this sentence would be excluded from the keyword-matching stage, which consequently prevents it from becoming a false positive match.

Exclusion Module – Non-Patient Individuals in Complex Sentences

For complex sentences, it was found that errors were being caused by the ambiguity arising from clauses formed over a direct object that was not the patient. In order to resolve this, if it was determined that if a sentence had the patient as a nominal subject but a non-patient as a direct object, then the algorithm checked to see if the

[9] The names of the dependency nodes are based on the Stanford typed dependencies manual [25].

sentence contained any education keywords. If any keywords were found, it would potentially be a positive match if it indicates the patient's current education level and required further processing.

To determine whether the education keyword was related to the patient or the direct object, a constituent tree of the sentence was first generated. The algorithm then searched the grammar productions of the tree to find a noun phrase (NP) consisting of a non-patient subject modified by a clause beginning with a subordinating conjunction node (SBAR). If a non-patient individual was not found at the head of the clause, then it was determined that any education keywords found within the clause would be related to the patient and the sentence was not excluded.

If the head of the clause was identified as a non-patient individual, the clause needed to be removed before coding can occur. For example, consider the following sentence and corresponding constituent tree in Figure 3:

"She has a **brother** who is a freshman in **college**."

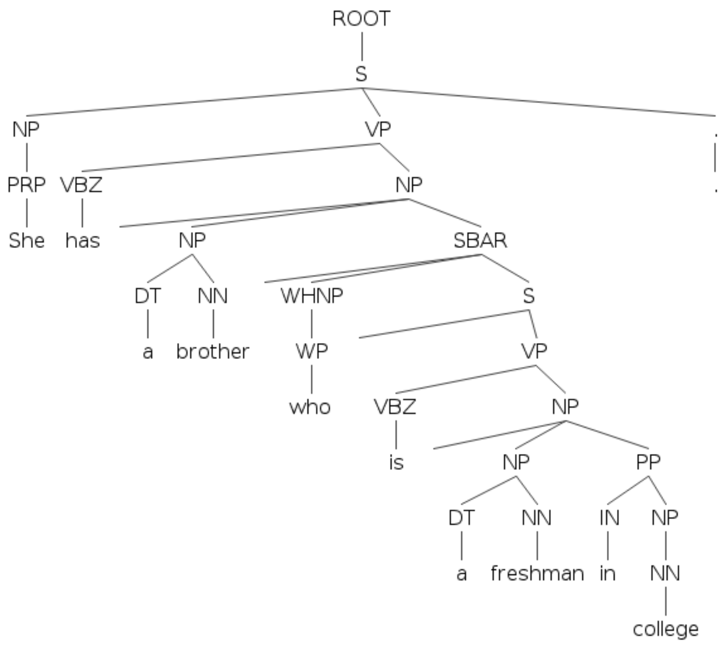

Figure 3. Graph of Constituent Tree Output from the Stanford Parser.[10]

The constituent tree is a positive match for an NP node followed by an SBAR node. The noun phrase "a brother" contains the embedded SBAR clause "who is a freshman in college" which contains the keyword "college". As a result, this clause was deleted from the sentence. The remaining portion of the sentence was then checked for keywords. If additional keywords were found, it was then passed on to the keyword matching stage for coding.

Exclusion Module – Speculative Sentences

The second type of error resolved by the exclusion module was sentences expressing future intent to attend school, which were not indicative of the patient's current education level since such sentences do not confirm that the patient had actually enrolled. For these sentences, attempts were made to develop rules based on

[10] This diagram was drawn using phpSyntaxTree (http://ironcreek.net/phpsyntaxtree/).

the constituent tree and the typed dependencies, but the processing provided by those tools were not deep enough to discern a generalizable pattern. Instead, a keyword search for the phrase “would like to” was used and if found, the algorithm determined whether it came before or after any target keywords for education level in the sentence. If a target keyword was found after the phrase “would like to”, then it was considered likely that it was related to a speculative phrase and the sentence was excluded. For example, the sentence “She would like to go on to graduate school” was excluded because the phrase “graduate school” was found after “would like to”.

Multiple Education Level Resolution Module

If a sentence was not excluded, it was then evaluated to determine if it contained keywords that indicated more than one education level, which was a source of coding error for the keyword-only version of the algorithm.[11] Sentences that indicated more than one education level were passed on to a NLP-based resolution module instead of the keyword matching stage. All other sentences were processed using the keyword matching stage as described in Section 4.2.1.

One technique explored to resolve multiple education levels mentioned in a sentence was to create a constituent tree of the target sentence. The sentence was then chunked into subunits, each corresponding to a verb phrase that was often connected with a conjunction. Further processing was then performed over each subunit separately. This proved to be effective on simple, well-formed sentences such

[11] Further details are provided in “Error Analysis” in Section 4.2.1.

“The patient does have one sibling and currently has been a student at ABC-COLLEGE and graduated from high school”, where the verb phrases “graduated from high school” and “been a student at ABC-COLLEGE” were chunked and processed separately at the keyword matching stage (Figure 4).

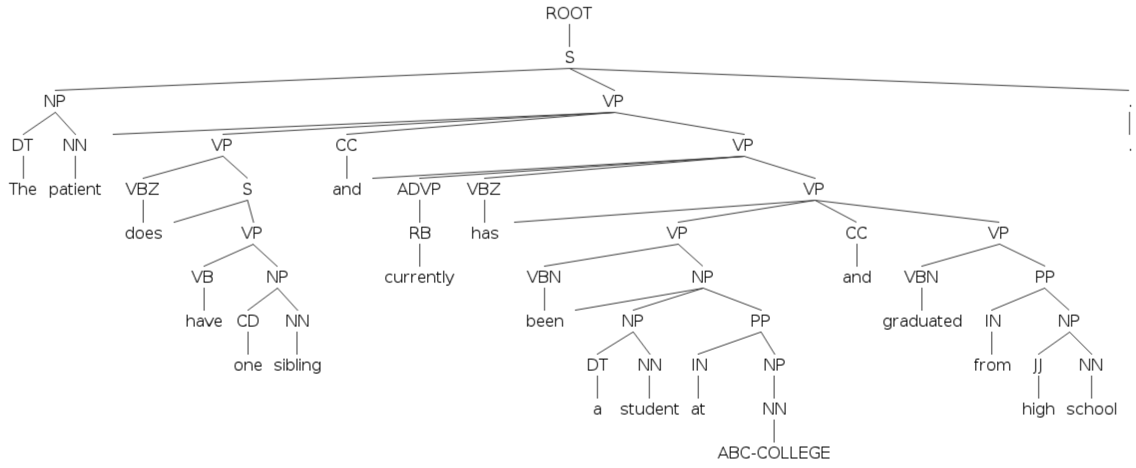

Figure 4. Constituent Tree Example of Chunking Using Verb Phrase Conjunctions.[12]

Not all sentences contained verb phrase conjunctions, such as “She attended graduate school recently with an undergraduate degree from ABC-COLLEGE.” Sentences such as this one required a different approach in order to examine the semantic and syntactic relations in the sentence at a deeper level. For sentences without verb phrase conjunctions, rules based on collapsed typed dependencies were developed.

Multiple Education Level Resolution Rule Type #1

The first set of rules targeted phrases indicating that the patient had attended school, meaning that they had at least some amount of that education level whether it is high school, college, or higher graduate education. These phrases will either match

[12] This diagram was drawn using phpSyntaxTree (http://ironcreek.net/phpsyntaxtree/).

for the “dobj” dependency if indicating that the patient attended school or match for the “prep_at” dependency if indicating that the student was at school:

> “She ….**attended college** at ABC-COLLEGE – where she met her husband” → dobj (attended-6, college -7)
>
> “The patient….currently has been a **student** at **ABC-COLLEGE**” → prep_at(student-7, ABC-COLLEGE-9)

<u>Multiple Education Level Resolution Rule Type #2</u>

The next set of rules were for phrases indicating that the patient had graduated from school. The algorithm searched for either the “prep_from” or “dobj” dependencies and whether it had either “graduate” or “graduated” as the first term in the tuple. If so, it then searched the second term in the tuple for the education level.

> “She **graduated** from **high school**” → prep_from(graduated-2, school-5); amod(school-5, high-4)
>
> “She **graduated high school**” → dobj(graduated-2, school-4); amod(school-4, high-3)

<u>Multiple Education Level Resolution Rule Type #3</u>

The third set of rules were for phrases indicating that the patient had received a degree or diploma and were also used to code a patient that had completed a particular education level. The dependencies were searched for a direct object (“dobj”) tuple containing one of a set of keywords such as “got”, “received,” “earned,” “has,” or “completed.” If matched, the algorithm then searches the tuple for further clarification to determine the code, whether by searching for specific degree names such as “BA,” “MBA,” or “Ph.D.” or by searching for the type of school using adjectival modifiers or noun compound modifiers. For example:

“The patient **has** her **college degree**” → dobj(has-3, degree-6); nn(degree-6, college-5)

A subset of rules checked whether the sentence indicated that a patient was actually still working on a degree, which for example would be coded as “Some College” in the case of the patient working on college diploma. These rules involved searching for a “prep_on” dependency containing the word “working” and then searching for modifiers to clarify the type of degree.

“She is **working** on her **GED**” → prep_on(working-3, GED-6)

Multiple Education Level Resolution Rule Type #4

Rules for negation detection were included in this particular set of rules. This involved having the algorithm search for a “neg” negation modifier dependency and checking to see if the first word in the “neg” tuple is also found as part of one of the previously mentioned dependency rules.[13]

“The patient….did **not graduate** from **high school**” → neg(graduate-5, not-4); prep_from(graduate-5, school-8); mod (school-8, high-7)

[13] A separate negation detection module similar to what was used here was also implemented in the keyword-only coding module to search for negations of sentences indicating graduation but did not require dependency parsing.

The final results of the algorithm on the development set of 110 documents are presented as an ablation study in Table 7, which occurred in multiple steps where an additional NLP-based rule is removed with each subsequent step. The complete algorithm with all NLP-based rules included showed very strong performance in all areas. By removing the verb phrase constituency rules (VPC removed), coding accuracy slightly increased. Analysis of the errors showed that one document had been incorrectly processed by the Stanford Parser, adding an extra space in the word "master's" so as to prevent a correct keyword match.

At the other steps in the ablation study, removing the negation detection rules throughout the algorithm as well (VPC + Neg removed) resulted in a decrease in coding accuracy. Removing the dependency parsing module for mixed education levels, leaving only the exclusion module and the keyword-only based matching algorithm still active (VPC + Neg + Dep removed), also decreased coding accuracy as well as text accuracy. Lastly, with the exclusion module removed (VPC + Neg + Dep + Exc removed), there was a decrease in coding accuracy, text accuracy, recall and precision. The combination of the exclusion module, dependency parsing rules for mixed education levels, and the negation detection rules provided the best results.

	Development Set (110 documents)				
	Code Accuracy	Text Accuracy	Recall	Precision	F-Score
Complete algorithm	0.927	0.945	0.915	0.789	0.847
VPC removed	0.936	0.945	0.915	0.789	0.847
VPC + Neg removed	0.918	0.945	0.915	0.789	0.847
VPC + Neg + Dep removed	0.891	0.936	0.915	0.789	0.847
VPC + Neg + Dep + Exc removed	0.882	0.918	0.947	0.730	0.824

Table 7: Development Set Performance of the NLP-Based Education Level Algorithm.

Documents that were still generating errors were doing so for several reasons and were set aside for future work. One main reason was due to complex sentences with multiple clauses containing keywords that indicate both finishing a particular education level and still currently attending at that level, which caused coding inaccuracies. For example, the sentence “He attended ABC-COLLEGE with a desire to get a degree in film” matched for the “Some College” classification due to the verb phrase “attended ABC-COLLEGE”. However, the sentence also matched for the phrase “with a degree” indicating the patient had completed schooling as it was more commonly found in sentences such as “The patient graduated from college with a Psychology degree”.

Another example is “She is currently attending ABC-COLLEGE in LOCATION where she has a degree in Spanish” which has the phrase “currently attending ABC-COLLEGE” to indicate “Some College” but also includes the phrase “has a degree in Spanish” which is typically used to indicate having already completed college and having the degree. In this case the use of the phrase “has a degree in Spanish” requires an alternate interpretation, instead meaning that she patient plans to earn a degree in Spanish.

Future work should focus on better discrimination within a particular education level between completing school and still currently attending.

Another unresolved error affecting recall and precision were ambiguous sentences that were annotated as being potentially helpful in determining the correct code by providing context to other sentences but were not as helpful on their own due to not matching for any specific keywords. For example, “He is currently on a leave of absence from school and unsure if he wants to go back to the same school” was annotated due to it being relevant to the patient’s education level despite not having any keywords indicating a specific kind of school on its own. As a result it was not counted as a positive match by the algorithm. Similarly, “She is in school in LOCATION and would like to be an economist” would be helpful in determining whether a patient is currently in some kind of schooling but a specific education level is not indicated. Additionally, the presence of the phrase “would like to” made it a candidate for exclusion by the exclusion module. Future work could work on coding based on context from multiple sentences at once, but not all of these ambiguous sentences are considered necessary to determine the correct code.

4.3 Results & Discussion

4.3.1 Phase 1 – Keyword Based Algorithm

The final version of the keyword-based algorithm in Phase 1 performed well on the last 90 documents in the corpus (code accuracy=0.74, text accuracy=0.84, recall=0.91, precision=0.77, F-score=0.84). There were some coding errors that were due to keyword variations in the text which were not present in the algorithm, but the algorithm still presented high recall and moderately high coding and text accuracy. Although the Social History section is typically noted as "SOCIAL HISTORY" in the assessments, one document had this section titled as "SOCIAL/DEVELOMPENTAL HISTORY" which caused an error in the section parser. Many of the other errors were similar to ones seen before that would be addressed with NLP, such as sentences that include references to multiple different education levels (in most cases high school and college) as well as sentences discussing a non-patient individual.

In general, the final version of the algorithm performed well over the test set during this phase. The high recall of the final version of the algorithm shows that the collection of keywords generalize well to this kind of document. This means that the majority of relevant sentences will be identified, although the keyword-based rules for coding are not quite as accurate. Precision is addressed in the second phase of algorithm development using NLP, which should also improve coding accuracy. The performance of the algorithm also showed that most documents can be given a code for education level by processing each sentence individually rather than requiring context from multiple sentences. Using the approach in which the code would be updated as the algorithm searches through each sentence with the assumption that the most recent education level would be expressed last also worked well.

4.3.2 Phase 2 – Addition of NLP-Based Rules

The algorithm with the addition of the NLP-based rules on the test set performed reasonably well based on the ablation study in Table 8. Results were the same between the step using the entire algorithm, and the step with the VP constituency and negation detection rules removed (VPC + Neg removed). When dependency parsing for multiple education levels was removed as well (VPC + Neg + Dep removed), code accuracy decreased. With the exclusion module removed (VPC + Neg + Dep + Exc removed), code and text accuracy decreased as well as precision, but there was a slight increase in recall. Although the exclusion module had decreased recall in both the development set and test set, the increase in precision was greater than the decrease in recall in both sets, showing that it is still beneficial in improving the accuracy of the algorithm. When compared to the performance of the algorithm over the development set in Table 7, there is a drop in overall performance of the algorithm over the test set. A baseline code accuracy calculated using random classification was found to be 0.28.

	Test Set (90 documents)				
	Code Accuracy	Text Accuracy	Recall	Precision	F-Score
Complete algorithm	0.778	0.856	0.9	0.808	0.851
VPC removed	0.778	0.856	0.9	0.808	0.851
VPC + Neg removed	0.778	0.856	0.9	0.808	0.851
VPC + Neg + Dep removed	0.756	0.856	0.9	0.808	0.851
VPC + Neg + Dep + Exc removed	0.744	0.844	0.914	0.771	0.837

Table 8: Test Set Performance of the NLP-Based Education Level Algorithm

Analysis of the errors in the test set indicated that there were additional ways of expressing levels of education that were not observed in the development set. There were also alternate ways that future plans of further education were expressed in the test set, such as “plans on getting her masters,” “wants to attend graduate school”, and “reports hopes to go onto medical school.” There was also one sentence that was missed by the negation detector indicating that the patient had no college degree because the negation detector was developed to only look for whether the patient had graduated or not. Again, these results were also affected by the one document had the “SOCIAL HISTORY” section noted as the “SOCIAL/DEVELOPMENTAL HISTORY” section and as a result was missed by the section parser due to a section header mismatch.

Other errors were due to sentences that were too complex for the rules based on the Stanford Parser output. One such example was “She has an older sister who is married with four children and a younger brother who is at home and goes to community college” which the algorithm did not exclude from coding and was used to determine the incorrect code. Although the algorithm should have chunked the phrase “a younger brother who is at home and goes to community college,” the phrase “and goes to community college” was not included due to it consisting of a coordinating conjunction and separate verb phrase.

Some sentences were also not well-formed enough for the parser to properly process, which can be a problem in clinical NLP. One such sentence was “She has an older sister age 22 currently attends ABC-COLLEGE, is in a sorority.” Because the sentence is not clear on whether “currently attends ABC-COLLEGE” is a relative clause about the older sister or is referring to the patient, it could not correctly determine whether to exclude the sentence or not. Since the parse did not detect a subordinating conjunction after the mention of an older sister, the sentence was not excluded and resulted in a coding error. Another instance was in a document that included the sentences "Has one brother age 26. Recently graduated from college." Again because it was not clear from the second sentence on its own who exactly graduated from college, it had not been annotated in the gold standard due to its ambiguity. Because the sentence was processed independently of any other sentences, it was a negative match for graduating from college however.

In general, the algorithm would have also benefited from more generalizable rules. This is evident in the drop in performance over the test set in comparison to the development set. The rules currently used in the algorithm are too reliant on lexical items which restrict the rules' ability to abstract to a wide variety of assessments. Future work should focus on utilizing patterns found in the structural representations found in the sentences. In hindsight, this would have made better use of the NLP-based approaches.

4.4 Conclusion

In conclusion, the evaluation of the algorithm's output of the test set shows that there was more variety to the documents than what was expressed in just the development set. As a result, more generalizable rules are necessary to increase performance rather than specific rules which were developed based on a limited number of documents. Despite the smaller size of the corpus however, the keyword-only algorithm set still demonstrated reasonable recall and the incorporation of NLP-based rules provided a measurable increase in performance.

Chapter 5: Marital Status Extraction and Coding

5.1 Introduction

This chapter describes the development and performance of an algorithm that automatically detects marital status in the psychiatric assessments and classifies each document with a code based on the identified text.   Section 5.2 describes the development of the algorithm in two phases, a keyword-based phase and a natural language processing-based phase. Like the education level algorithm, the goal of this part of the study was to determine if a keyword-only based approach would provide an acceptable baseline, and then if using NLP-based rules would result in a noticeable improvement. The algorithm's performance over the final test set is evaluated for both phases in Section 5.3.

Marital status is a demographic that is found in the free-text/narrative section of the psychiatric assessment in addition to education level. It is collected as a part of outcomes studies in order to contribute to demographic profiles of patients involved in research studies. Marital status is currently manually abstracted by outcome studies staff, assigned the most relevant code, and that code is stored in a database. The possible classifications for marital status include:

Single – This is used to indicate patients that have never married.

Married – This is used to indicate patients are currently married or re-married. This classification is also used for patients who are married but legally separated.

Divorced – This is used to indicate patients that are currently divorced.

Widowed – This is used to indicate patients that are currently widowed.

Unspecified – This is used to indicate that the patient's marital status could not be clearly determined from the psychiatric assessment.

5.2 Method

5.2.1 Phase 1 – Keyword-Based Algorithm

The development approach of this algorithm was the same as the one used to develop the algorithm to extract education level. The primary goal of the phase was to develop an algorithm that would use keyword-based pattern matching using regular expressions in order to identify sentences that indicated the patient's marital status and maximize the recall of those patterns. Keyword-based rules were also used to assign codes as well in order to determine whether it would be sufficient to use only keywords for coding or if more sophisticated approaches were necessary. Similarly, an iterative process was also used to develop this algorithm, beginning with a development set of 30 documents and a dev-test set of 20 documents, which were incorporated into the development set with each subsequent iteration.[14]

Each iteration of development was fairly straightforward, with additional keywords and regular expressions being added with each iteration based on the previous dev-test data. One challenge involved adding contextual keywords in order to

[14] For more information on this approach, see "Phase 1 – Keyword Based Algorithm" in Section 4.2.

disambiguate salient words. For example, the word “relationship” did not always mean a romantic relationship, as the Social History section may also describe other kinds of relationships between the patient and family members or friends. As a result the word “relationship” as a keyword required additional contextual keywords to increase precision, such as “significant”, “intimate”, and “romantic”.

Additionally, we used the presence of negation of the word “relationship” to make the downward-entailing inference that if no relationship exists, then no romantic relationship exists. This was especially useful for examples like the following, where the word “relationship” was mentioned without any specification of the nature of the relationship:

> “The patient has no current relationship and has no desire to be in a relationship, primarily related to self-assessment of negative body image relative to body dysmorphic disorder.”

Sentences that merely had the word “relationship” without any of the additional modifiers were not considered a positive match due to their ambiguity.

The set of regular expressions used in the final version of the algorithm are presented in Table 9.

Single	((significant, intimate, romantic)\srelationship(s?)), dating, going out, seeing, single, girlfriend, boyfriend, (no(t?)(.*)relationship(s?)), relationship(.*)none currently
Married	married, marriage, husband, wife, spouse
Divorced	was married, divorce(d*), divorcee, ex(-)*husband, ex(-)*wife, ended(.*)marriage
Widowed	((husband, wife)(.*)(passed away, died, deceased)), widow(ed)*

Table 9: Regular Expressions Used to Extract Marital Status

The codes were ranked in such a way that a higher ranked code would take precedence if a sentence matched one of the related keywords, similar to what was done for education level. The rank from lowest to highest was Single, Married, Divorced, and lastly Widowed. Figure 5 shows a flow chart of the algorithm's overall process, which is the same as the one found in the keyword-based education level extraction algorithm.

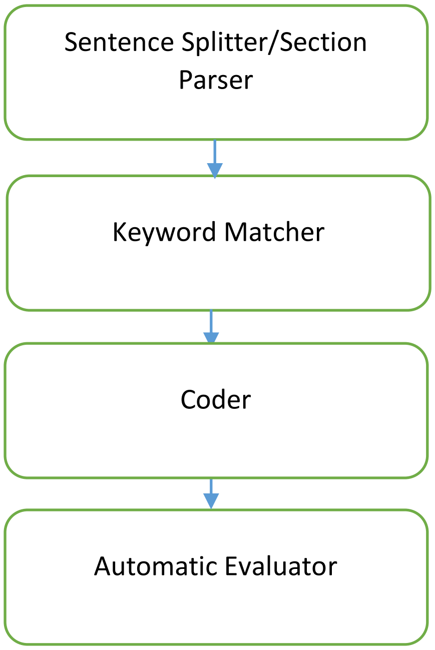

Figure 5. High-Level Abstraction of the Keyword-Based Marital Status Extraction Algorithm.

Results of using the dev-test set at each iteration are presented in Table 10. As discussed in Section 3.3, recall, precision, and F-score reflect the performance of the keyword-matching stage before coding occurs. Code and text accuracy reflect the algorithm's ability to determine the correct code from the correct text.

Iteration	Development Set Size	Dev-Test Set Size	Code Accuracy	Text Accuracy	Recall	Precision	F-Score
1	30	20	0.80	0.80	0.92	0.73	0.81
2	50	20	0.65	0.7	0.82	0.56	0.67
3	70	20	0.90	0.90	1.0	0.67	0.80
4	90	20	0.90	0.90	0.95	0.78	0.85

Table 10: Dev-Test Set Performance of the Keyword-Based Marital Status Algorithm.

Across all of the development iterations (1 through 4), code and text accuracy quickly plateaued at 0.90 although with a small decrease in the second iteration. Recall also experienced a decrease in the second iteration but quickly increased to the 0.95-1.0 range. Recall of the algorithm before the final evaluation was not perfect due to a few sentences that could be interpreted as indicating marital status but were too vague to contain any specific keywords that could be included. The recall was also affected by some formatting errors introduced by the annotation tool which caused a small number of instances to be missed.

Error Analysis

During development, it soon became obvious that one of most common errors in each iteration that would have to be addressed with NLP-based rules were due to sentences expressing the marital status of people who were not the patient. This was adversely affecting precision and consequently also lowering code and text accuracy. For example:

> “The patient reports his parents have been married for 20 to 25 years and have a happy marriage.”

These types of sentences were considered a positive match for marital status keywords, which resulted in a large amount of false positives that decreased precision. It also decreased code and text accuracy since these sentences were then used to determine the document code.

Another error was based on the issue regarding a sentence that indirectly mentioned a keyword related to marital status but did not clearly expressed any marital status. The following example illustrates this:

> “The patient did attend an all girls school and reported significant anxiety about dating; although, she does have a core group of friends per patient and mother's report.”

Although some may interpret this sentence as the patient being single, it actually only discusses the patient’s ability (or inability) to form close relationships with others, which were not annotated in the gold standard. This contributed to the decrease in precision, code and text accuracy as well.

In the next phase, NLP-based rules will be used in an attempt to increase precision and consequently increase code and text accuracy in the test set by addressing the aforementioned errors identified in the development set.

5.2.2 Phase 2 – Addition of NLP-Based Rules

In the second phase of this study, NLP-based rules were used to resolve the errors found in the development set after the final keyword-only version of the algorithm was developed. In order to increase precision, an exclusion module was developed in order to detect sentences that contained subjects that were not the patient and exclude them from the keyword-matching stage. The second revision to the algorithm was the addition of a negation detection module that would more specifically target ways that a patient either is not currently in a romantic relationship. The flow chart depicting the revised process is shown in Figure 6.

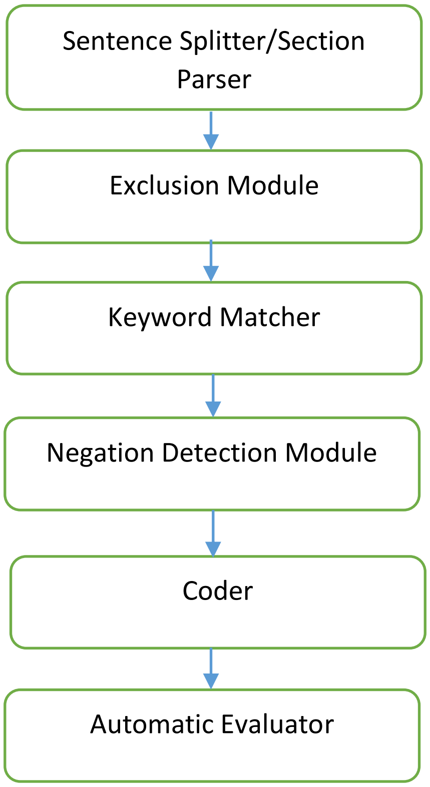

Figure 6. High-Level Abstraction of the NLP-Based Marital Status Extraction Algorithm

The exclusion module used to determine who the marital status was being predicated about was based on the one used in the algorithm to extract education level. The same rule was implemented in which the nominal subject was identified using typed dependencies. If the nominal subject matches for any keywords indicating a non-patient individual, it is excluded. If a nominal subject matches for one of the keywords used to indicate the patient then it is not excluded. The keywords used to indicate patients and non-patients are presented in Table 11.

Patients	he, she, patient
Non-patients	mother, father, brother, sister, sibling, parent, cousin, aunt, uncle, they

Table 11: Patient and Non-Patient Keywords

The pronouns “he” and “she” included in Table 11 are always assumed to be referring to the patient if it is a nominal subject. Although this is an understandably large assumption, it is considered appropriate for clinical texts where the patient is the focus of the narratives. Consequently, the patient is more likely to be referenced as a singular pronoun than non-patient individuals. Lastly, additional rules used in the exclusion module for education level which were based on constituency parses were left out of this version. This is because the errors those rules aimed to resolve were not observed in the sentences discussing marital status and the rules themselves were decreasing the performance of the algorithm.

The second NLP-based approach implemented in this phase was an algorithm that would more precisely detect keywords related to marital status that were negated.

This module was developed due to concerns over the broadness of the regular expression used for negation detection in the keyword-only algorithm, which was *(no(t?)(.*)relationship(s?))*. The problem with that regular expression is that the algorithm does not look for a semantic connection between the terms used for negation ("no" and "not") and the keyword "relationships". Although there were not any errors in the keyword-only stage of algorithm development based on the use of this regular expression, it was proactively developed in case of potential errors in the test set.

The negation detector uses the Stanford Parser's collapsed typed dependency output to identify negation patterns in a sentence. It was included in the algorithm after the keyword matching stage and its only goal is to increase coding accuracy. All of the dependency-based rules were used to classify the patient as "Single."

Marital Status Negation Rule Type #1

The first type of rule used focused on sentences that indicate that the patient had no relationships or dating history. It primarily made use of the "neg" dependency type and how it affects nouns related to marital status. For example:

> "**No** current significant **relationships** although she does describe significant friendships." → neg(relationships-4, no-1)

A variation of this rule searches for the term "dating history", which was commonly found in the corpus, being negated:

“Does appear to be somewhat socially avoidant; has **no** significant **dating history**.” → neg(history-13, no-10); amod(history-13, dating-12)

Marital Status Negation Rule Type #2

An additional set of related rules target sentences that express that the patient is not in a relationship or dating. It also makes use of the “neg” dependency type but also uses the “prep_in” dependency type and focuses on how verb phrases such as “in a relationship” or “is dating” are negated:

“She **is not** currently **in** a romantic **relationship**.” → neg(is-2, not-3); prep_in(is-2, relationship-8)

Marital Status Negation Rule Type #3

Other rules were developed for negative content words which convey a particular concept while also incorporating negation. This requires additional rules that search for the direct object being modified in the verb phrase with the verb “denies” as the head. After evaluation of the algorithm using the test set it was determined that there were additional words and phrases which also serve this function besides the words “denies”, and at this point this rule may be considered to be too specific.[15]

“He does not report significant friendships and **denies** any serious **dating** or romantic **relationships**.” → dobj(denies-8, dating-11); dobj(denies-8, relationships-14)

[15] This is discussed in greater detail in the error analysis in Section 5.3.2.

Marital Status Negation Rule Type #4

Lastly, rules were added to search for the word "never" and whether it is modifying a keyword for marital status. Again, after final evaluation of the algorithm this rule could be considered to be too specific:

"She has **never** been **married** and has no children." → neg(married-5, never-3)

The effects of the NLP rules in the algorithm on the development set are illustrated in the results of an ablation study in Table 12.

	Development Set (110 documents)				
	Code Accuracy	Text Accuracy	Recall	Precision	F-Score
Complete algorithm	0.972	0.964	0.925	0.902	0.914
Neg removed	0.964	0.964	0.925	0.902	0.914
Exc and Neg removed	0.827	0.827	0.963	0.713	0.819

Table 12: Development Set Performance of the NLP-Based Marital Status Algorithm

Removing the negation detection module slightly decreased coding accuracy, due to the sentences expressing that the patient was never married were no longer properly being processed. Removing the exclusion module greatly decreased precision as well as code and text accuracy, showing that it was having a positive effect on the output despite a decrease in recall.

5.3 Results & Discussion

5.3.1 Phase 1 – Keyword-Based Algorithm

The algorithm performed well over the test set (code accuracy=0.73, text accuracy = 0.80, recall = 0.93, precision = 0.62, F-score = 0.74). The high recall indicates that the keywords cover a broad spectrum of sentences that indicate marital status. Code and text accuracy were also acceptable, although not as high as the results in the development iterations. This may be partly attributed to the moderately low precision, which is resulting in many false positive sentences being matched by the algorithm and being used to determine the code. These false positives were primarily the result of sentences containing one or more keywords related to marital status but were referring to a family member's marital status. This source of error is addressed in the second phase through the use of natural language processing and an increase in precision is expected.

5.3.2 Phase 2 – Addition of NLP-Based Rules

The results of the ablation study on the test set are presented in Table 13.

	Test Set (90 documents)				
	Code Accuracy	Text Accuracy	Recall	Precision	F-Score
Complete algorithm	0.889	0.878	0.907	0.845	0.875
Neg removed	0.867	0.933	0.907	0.845	0.875
Exc and Neg removed	0.733	0.8	0.926	0.617	0.741

Table 13: Test Set Performance of the NLP-Based Marital Status Algorithm

The complete algorithm showed strong performance, with all metrics in the satisfactory range. With the negation detection module removed (Neg removed), there was an increase in text accuracy although code accuracy decreased. With both the exclusion module and negation detection removed (Exc and Neg removed), there was a significant drop in code and text accuracy as well as in precision and F-score. There is also a noticeable drop in the overall performance of the algorithm over the test set when compared to performance over the development set in Table 12. A baseline code accuracy calculated using random classification was found to be 0.34.

The increase in text accuracy after negation detection was removed could be attributed to the negation detection module not matching for variants in sentences expressing negation that were not observed in the development set. The lowered performance of the algorithm over the test set could also be attributed to this. One common variant found in several sentences was negation expressed in present and past perfect verb phrases. Examples include:

> "Has not had real significant close dating relationships."
>
> "Had been married for 17 years and has three children."

Such sentences were not present in the development set. Other sentences indicating negation in the test set were not well-formed sentences which the algorithm did not have negation rules to account for. These sentences were brief descriptive shorthand phrases that can appear in clinical notes and is a common challenge in clinical NLP. For example:

> "No history dating."
>
> "Again, no dating."

Although additional rules could have been created if these kinds of sentences were observed in a larger development set, a more effective approach would have been to abstract the current rules by taking greater advantage of patterns found in the structural representations of the sentences. Currently, the incorporation of individual lexical items identified only in the development set limit the dependency rules by making them too specific. In order to more effectively utilize NLP-based approaches, the use of lexical items should be minimized.

Either the constituency parses, dependency structures, or a combination of the two could have been used to a greater extent in order to further improve performance. For example, “No history dating” could have been matched with a rule that identified any kind of modifier for the word “dating” including verbal modifiers (by using the regular expression **mod*), rather than just adjectival modifiers. Another example would be allowing for a variety of keywords to be negated rather than just “relationship” and “dating history”. This would have resolved the sentence “Again, no dating.” Lastly, constituency trees could have helped with resolving “Has not had real significant close dating relationships” by identifying verbs in a verb phrase containing keywords related to marital status and then determining if that verb is being negated. Examples such as these should be a primary focus for future work.

Another kind of error was involved with additional ways that marital status was being expressed in the test set that was missed by the algorithm. One pattern was sentences indicating that the patient had recently ended a relationship, which was annotated as “Single”. Examples include:

> “She has had a long term relationship which broke up in the last two to three months.”

“She has been in a long term 5 year relationship which is currently on hold.”

Adding additional keywords such as “broke up” or “on hold” as negative content words would be beneficial in identifying these negations. Because the dependency rule that utilized negative content words only included the keyword “denied”, it was too specific to identify other words and phrases that perform a similar function. Future work should aim to develop a more comprehensive list of negative content words in order to improve generalizability.

5.4 Conclusion

Despite the remaining errors, the reasonably high results indicate that the algorithm can correctly code a majority of the documents in the corpus. The keyword-only approach provided an acceptable baseline and the addition of NLP-based rules showed a dramatic increase in accuracy and precision. The noticeable decrease in performance of the algorithm over the test set in comparison to the development set indicates that the rules for negation detection are too focused on specific lexical items rather than broader structural representations provided by constituency trees and dependency parses. Minimizing the influence of lexical items on these rules are expected to improve performance in future versions of the algorithm.

Chapter 6: Diagnosis, Age, and Gender Extraction

6.1 Introduction

In this chapter, the development of three separate algorithms to extract and code the patient's age, gender and their first five current admission diagnoses will be discussed. Diagnosis information is very useful piece of clinical data that can be used to classify patients for suitability for clinical trials or other research studies. Abstracting diagnoses into codes allows for easier comparisons between groups of patients based on diagnosis. Similarly, age and gender are also important demographics to collect for research purposes as well.

Because these three categories are found in a more structured format in the assessments, it was not anticipated that anything beyond a keyword-based approach would be necessary. As a result, these algorithms are based on simpler methods than the algorithms used to extract education level and marital status. Additionally, we expect to see better results over both development and test sets.

6.2 Diagnosis Extraction

6.2.1 Method

Codes are assigned to the first five diagnoses given to a patient upon admission by a board-certified psychiatrist at Rogers Memorial Hospital. The vast majority of the documents in the corpus had diagnosis information located in a table towards the end of the assessment.[16] The algorithm only targeted Axis I diagnoses, which are the clinical psychiatric disorders as described in the 4th edition of the *Diagnostic and Statistical Manual of Mental Disorders* [33]. A group of 28 diagnosis codes were used for this study and were based on the most frequently occurring diagnoses as observed by hospital staff who currently manually code these documents.

A section parser was first used to extract the "Axis I" section of the text. We observed that individual diagnoses were often delimited by either a period or a number enclosed in parentheses, and these delimiters were used to identify the text strings corresponding to individual diagnoses.[17] Each text string was then individually processed through an exclusion module and a preprocessing module for depression diagnoses. These modules were developed in order to handle special cases that could not be coded using just the diagnosis terms themselves.

The exclusion module searched for diagnoses containing keywords that indicate whether the diagnosis is either ruled out, not conclusive, in complete remission or by history. For example:

"**Possible** obsessive-compulsive disorder."

This would be excluded from coding because it was not a confirmed diagnosis.

[16] For an example of a diagnosis table found in the corpus, please see Table 1 in Section 3.1.

[17] Backslashes, parentheses, numeric digits, and periods were also stripped from the text strings. Commas and semi-colons were allowed however because they are used to separate the main diagnosis from a specifier (e.g. "Generalized anxiety disorder, rule out social anxiety").

A challenge encountered while developing this module was determining whether one of these keywords was describing the main diagnosis or an additional specifier included on the same line. A specifier is typically a clarification given to a diagnosis in order to rule out similar diagnoses or to describe more specific features of the diagnosis. They are usually separated from the main diagnosis with a delimiter such as a semi-colon. The following example shows the specifier in bold:

> "Eating disorder, not otherwise specified; **rule out anorexia nervosa binge/purge type**."

The text in this example would be allowed to pass through to the coding module because the phrase "rule out" was found in the second text segment after the semi-colon delimiter.

If the text passed through the exclusion module, the algorithm then checked to see if the diagnosis indicated either "depression" or "major depression." If the text matched for "major depression", it passed through unchanged. If it matched for only "depression", the term was mapped to the concept name "mood disorder not otherwise specified" which is a broad classification given in cases where the criteria for major depression are not met. This was because the term "depression" on its own is not found as a diagnosis in the DSM-IV and "mood disorder not otherwise specified" is the nearest analogous concept.

After passing through the exclusion and preprocessing module the text is then searched using a dictionary of regular expressions containing the diagnoses and their associated codes. If the text was a match for the regular expression, it then was then assigned the related code. If the text string was not a positive match for any of the regular expressions in the dictionary, then it was assigned the code for “Other”.

If a text string was not present for one of the first five diagnoses, then the classification “No Diagnosis Present” was assigned. Because no text string was found in these cases, they would not count towards recall, precision, and F-score but did contribute to code and text accuracy[18]. The regular expressions used were first based on the DSM-IV vocabulary extracted from the UMLS Metathesaurus. Due to the variety of ways that the diagnoses were expressed in the corpus however, the keywords required additional revisions beyond what was given in the Metathesaurus based on empirical observation of the documents in the development set. For example, the Metathesaurus did not include separate terms for the Restricting and Binge-Purge subtypes for anorexia nervosa.

The results of the final algorithm on the development set are presented in Table 14. The term “Primary Diagnosis” indicates that first current diagnosis found in the “Axis I” part of the diagnosis table that was considered eligible for coding. Because the corpus used in this study came from an eating disorders unit, this was typically found to be an eating disorder diagnosis. “Secondary Diagnosis” was considered to be the second current diagnosis found in this section that was considered for eligible for

[18] Text accuracy was counted for documents coded as “No Diagnosis Present” in order to align with code accuracy, indicating that the coding of “No Diagnosis Present” is correct due to the absence of text. Since recall, precision, and F-score indicates the accuracy of the keyword-matching ability of the algorithm, it was not counted for those metrics.

coding. Similarly, "Tertiary Diagnosis was the third current diagnosis, "Quaternary Diagnosis" was the fourth current diagnosis, and "Quinary Diagnosis" was the fifth current diagnosis.

	Development Set (110 documents)				
	Code Accuracy	Text Accuracy	Recall	Precision	F-Score
Primary Diagnosis	0.991	0.982	0.982	0.991	0.986
Secondary Diagnosis	0.973	0.973	0.970	0.990	0.980
Tertiary Diagnosis	0.982	0.982	0.968	1.0	0.984
Quaternary Diagnosis	0.991	1.0	1.0	1.0	1.0
Quinary Diagnosis	1.0	1.0	1.0	1.0	1.0

Table 14: Development Set Performance of the Diagnosis Algorithm

The remaining errors in the development set were caused by the sentence splitter, which either failed to separate individual diagnoses into separate text string units according to the delimiters, or included the "Axis II" heading on the same line as an Axis I diagnosis, which caused problems with the section parser as well as the regular expressions in the coding algorithm. The results show that the sentence splitter was functioning sufficiently for the majority of the documents however.

6.2.2 Results & Discussion

The results on the test set are presented in Table 15.

	Test Set (90 documents)				
	Code Accuracy	Text Accuracy	Recall	Precision	F-Score
Primary Diagnosis	0.922	0.978	0.978	0.989	0.983
Secondary Diagnosis	0.867	0.9	0.892	0.914	0.902
Tertiary Diagnosis	0.867	0.867	0.845	0.882	0.863
Quaternary Diagnosis	0.856	0.911	0.815	0.785	0.800
Quinary Diagnosis	0.944	0.956	0.778	0.700	0.737

Table 15: Test Set Performance of the Diagnosis Algorithm

Performance of the algorithm on the test set was satisfactory. The primary source of error in the test set was due to errors in the sentence splitter, which did not always properly split the diagnoses into distinct units. This error had a cumulative effect because if the primary and secondary diagnoses were not split into separate text strings, the tertiary diagnosis was coded as the secondary diagnosis, the quaternary diagnosis was coded as the tertiary diagnosis, and so on. This is evidenced by the decrease in code accuracy and text accuracy from the primary through the tertiary diagnoses and the constant decrease in F-score throughout all of the diagnoses. To increase performance, future versions of the algorithm should utilize an improved sentence splitter to reduce errors. It is speculated that code and text accuracy increased in the quaternary and quinary diagnoses due to the fewer number of documents that had more than three diagnoses to code. This resulted in an increase in "No Diagnosis" codes which are simpler for the algorithm to detect. Recall and precision continued to decline however.

Random classification was used to calculate a code accuracy baseline for each of the diagnosis categories. The baselines were as follows: primary diagnosis = 0.23, secondary diagnosis = 0.19, tertiary diagnosis = 0.14, quaternary diagnosis = 0.59, quinary diagnosis = 0.89. The random classification baselines were higher in the quaternary and quinary diagnoses due to fewer coded diagnoses present in the development set for those categories. The rule-based algorithm showed a higher code accuracy for each of the diagnosis categories in comparison to random classification.

Overall, the algorithm was successfully able to identify the majority of psychiatric diagnoses present in the corpus and assign the appropriate code. Although arbitrary codes only relevant to research at one specific hospital were used for this algorithm, the algorithm could potentially be used to facilitate mapping of psychiatric diagnoses found in the corpus to other controlled classification terminologies containing DSM-IV diagnoses such as ICD-9 or ICD-10. However, the Metathesaurus was not considered comprehensive enough on its own to achieve the specificity required for this task and additional modules had to be developed to handle special cases containing specifiers. Because of the wide variety of ways that psychiatric disorders could be expressed which required modifications to the terminological resource and additional pre-processing modules, more sophisticated text mining techniques such as machine learning may be more effective for this task and should be explored.

6.3 Age Extraction

6.3.1 Method

Age was expressed in a common pattern throughout the documents as a number (either presented as numerals or spelled out in words such as “twenty-three”) followed by a variation of the phrase “years old”. For example,

> “The patient is a **27-year-old** who has been in active treatment for anxiety and depression.”

This allowed for a simple pattern recognition algorithm to be developed that could search for variants of the phrase “year old” and extract the adjacent age. Ages spelled out as words were converted into numerical format. In order to reduce the likelihood of extracting the age of someone other than the patient, the section parser was used to restrict the algorithm's search scope to the Chief Complaint, History of Present Illness, Identifying Information and Impression sections.

6.3.2 Results & Discussion

Because of the more straightforward nature of this algorithm, an iterative development approach was not necessary, and the algorithm was developed using all 110 documents in the test set at once. The final algorithm over the development set resulted in a code and text accuracy of 1.0, a recall of 0.995, a precision of 0.972, and an F-score of 0.984. The results over the test set were a code and text accuracy of 1.0, a recall of 0.994, a precision of 1.0, and an F-score of 0.997. These results demonstrate that the algorithm is very reliable in extracting the age of the patient even from unstructured text.

6.4 Gender Extraction

6.4.1 Method

Gender is a simple but crucial demographic and was found in a structured format in the corpus. In order to ensure that the indication of gender was related to the patient, gender was extracted in the section with the heading of "SEX", which always expressed only the patient's gender. The algorithm was developed using all 110 documents in the development set at once. The automatic evaluator was only used for evaluating the code for gender while all other text-related metrics were calculated using manual evaluation. This is due to errors found in the automatic evaluation caused by a difference between how the original text was annotated and how the sentence splitter processed this section.

6.4.2 Results & Discussion

The algorithm performed well over the development set, with a code and text accuracy of 1.0, as well as a recall, precision, and F-score of 1.0. The test set yielded a code and text accuracy of 0.989, a recall of 0.989, a precision of 1.0, and an F-score of 0.989. A baseline code accuracy calculated using random classification was found to be 0.61. The only cause of error in the test set was due to the sentence splitter, which subsequently caused an error with the section parser for one document. These results show that the algorithm is well-suited to extracting this demographic in a well-structured format.

6.5 Conclusion

As expected, all three algorithms' performances over the development set were very good and provided a strong baseline for comparison to the test set. The results using the test set were very accurate for extracting age and gender and also satisfactory for diagnosis. Because the sentence splitter was a source of error for the diagnosis and gender extraction algorithms, it may not generalize well to more structured formats that do not contain proper sentences. Future work should include using a sentence splitter that is more suited towards the sections containing gender and diagnosis information. Additionally, the use of terminological resources in developing psychiatric diagnosis extraction algorithms should be evaluated in the future due to the limitations found in the Metathesaurus. Alternatively, more sophisticated text mining approaches may need to be considered given the variety of diagnosis terms and specifiers found in the current corpus.

Chapter 7: General Discussion and Conclusion

7.1 Discussion

The first question that this study hoped to answer was whether rule-based algorithms could accurately identify text that expresses a given category and assign the correct code. The results of the algorithms' performance over the test set indicates that each algorithm was able to perform this task successfully with a reasonable number of rules. This also shows that the majority of text that expresses demographic information can be considered to be fairly homogenous in lexical form and grammatical structure.

The second question was whether keyword-based pattern matching rules were sufficient for these text mining tasks, or if rules based on natural language processing were necessary. Although the algorithms used to extract and code age, gender, and diagnosis did not require additional NLP-based rules, the algorithms for marital status and education level did require further processing using constituent trees and typed dependencies. This was to be expected due to those two types of information being found exclusively in the narrative section of the documents. Although clinical narratives can be known contain poor sentence structure and abbreviated words which can reduce the effectiveness of natural language processing approaches, the corpus used in this study had the benefit of containing mostly well-formed sentences. This shows that accurate information extraction is possible given a well-written clinical narrative.

The final research question was if any aspects of a given algorithm could be generalizable to another algorithm. It was found that the exclusion module that identified subjects and heads of clauses that were not the patient and excluded them consideration for coding was useful in both the education level and marital status extraction algorithms. This may also be useful in rule-based extractions of other kinds of demographics using clinical texts and could also be modified to exclude sentences that are about the patient instead in order to extract data such as family history. The part of the exclusion module that identifies certain phrase structures such as relative clauses could also be useful given a larger corpus that has wider variety of sentences to use for development.

Apart from automatic coding, another possible application of these algorithms would be to help with developing structured data entry forms for demographics and diagnoses. The developed rules and lists of keywords could be used as part of an analysis of psychiatric evaluations in order to determine what kinds of structured formats to include in the data entry form as a data-value set pair. These algorithms can help determine what kinds of contexts should be considered when developing additional classifications. For example, gender can require context in order to more accurately determine a patient's gender identity. As a result, certain keywords could be identified to determine additional gender classifications such as intersex, transgender, or unspecified.

7.2 Limitations

Formatting Inconsistencies

One limitation observed throughout this study was that there were inconsistencies in how each version of the psychiatric assessments was formatted, which affected the precision of the automatic evaluator when comparing the extracted text to the gold standard. The initial plan was that the automatic evaluator would be able to identify if the text extraction algorithm determined the code from the correct text by comparing the text offsets between them; if the text offset of the extracted text intersected with the text offset of the annotated text, it would be considered a positive match.

However, the documents had to go through two stages of conversion before being processed by the algorithm, which introduced additional carriage return and newline characters at each stage. The first conversion stage was from the original Microsoft Word file accessed in the electronic medical record to a generic text file (.txt format). The second conversion stage was during the processing through the sentence splitter. Because the gold standard was based on the generic text file before it was processed by the sentence splitter, the additional characters resulted in an inconsistency between the text offsets. In hindsight it would have been wiser to have stripped all extra white space, newline characters, and carriage return characters from the generic text file before it was annotated to help reduce these inconsistencies. Although the automatic evaluator based on string-matching that was used instead was still accurate enough to complete the study, comparing text offsets would be more precise and is suggested for future studies.

Corpus Size and Generalizability within the Corpus

Another obvious limitation was the small size of the corpus. Because each document had to be manually de-identified as well as due to concerns about providing access to a large amount of sensitive protected health information, only 200 documents were allowed for this study. Because the corpus had to be separated into a development and a test set, this reduced the amount of documents to develop the algorithm with even more. Although the performance of all of the algorithms indicate that there was enough variety in the development set to generalize the algorithm to a reasonable level, there were still additional errors found in the final test set for some of the algorithms that were not present in the development set which could have been used to further improve performance. Future versions of the diagnosis, marital status, and education level extraction and coding algorithms would especially benefit from a larger corpus for development.

What would have improved the algorithm even more beyond a larger corpus as previously discussed is increasing the generalizability of the NLP-based rules used to extract education level and marital status, especially in regards to rules utilizing dependency parsing. After reflecting on the overall results of the study it can be better understood that the goal of NLP is to move beyond the specific lexical items found in a given sentence and to analyze the structure of the sentence instead. During the development of the algorithms, too much focus was placed on identifying syntactic and semantic relations between specific keywords. Although they did help resolve errors that were found in the development set, it had to be assumed that the demographic information would be expressed in a very similar way in the test set using the same keywords. The difference in results between the development and test sets for marital status and education level indicate that this is not the case. Future work should aim to develop rules based on higher-level sentence structures that are not limited as much by lexical items, which should improve generalizability and overall performance.

Generalizability to Other Clinical Documents

There are also limitations in regards to the algorithm's generalizability to other clinical documents. All of the algorithms used in this study were developed based on a corpus consisting of assessments from only one hospital. These documents were considered suitable for this study because they were well-structured in that they had consistently defined headings between each section. This allowed for simpler section parsing. The narrative sections also consisted of primarily well-formed sentences that would suit natural language processing techniques. Although these strengths allowed the algorithm to perform generally well over the corpus, one cannot assume that psychiatric assessments from another hospital or even from a different psychiatrist at the same hospital will provide similar results. The section parser does limit the algorithm's flexibility in identifying relevant text throughout a document and the algorithm is also reliant on well-formed sentences that are simple to analyze. The algorithms' ability to generalize well to psychiatric assessments from variety of sources should be a goal for future work.

Limitations of Terminological Resources

The UMLS Metathesaurus, which was used to develop the diagnosis extraction algorithm, was not considered comprehensive enough in regards to its DSM-IV diagnosis terms. Some subtypes and specifiers were not included at all, and other diagnosis terms contained specifiers that could not be generalized to cover the wide variety of ways that specifiers are expressed. As a result, the Metathesaurus provided a useful starting point for collecting potential terms to use for keyword matching but additional modifications and mappings needed to be developed beyond what the Metathesaurus provided. This indicates that alternative terminological resources may be needed to expand the amount of diagnosis terms covered by the algorithm in the future, or that more sophisticated approaches such as machine learning may be more suitable.

Anaphoric Expressions

The education level and marital status algorithms did not address the concern of properly resolving anaphoric expressions. Instead, the algorithms had made the assumption that anaphoric expressions would always be referring to the patient. This was because the documents used in the corpus were primarily focused on the patient as the subject. Although this assumption did not result in any errors in either algorithm, there is a chance of such an error occurring in the future. As a result, there should be further consideration in properly resolving anaphoric expressions with their proper antecedent or postcedent.

7.3 Conclusion

Based on the results of this study, it could be considered possible to eventually implement an automatic method of extracting demographic information and diagnoses from clinical text and assign them a code using a rule-based approach. Table 2 in Chapter 3 illustrates the degree of human accuracy in coding each information type, and these results provide a benchmark for the automatic extraction method to reach in future work. Although extracting demographics was initially considered to be a simple task, it can be more difficult than expected depending on the type of information one is looking to extract. Although some information can be easily extracted from clinical documents if it is in a structured format, there are still some unexpected challenges present in making sure that the document is properly formatted beforehand and that text strings are properly separated.

Other types of information that are only available in clinical narratives present an even greater challenge. It must be ensured that the information is the most currently available given what is provided in the document, that the information is actually referring to the patient, and that the information can be abstracted into the correct code. Considering that these obstacles were common across several different types of information extracted during this study, it is safe to presume that they should also be kept in mind when extracting any kind of demographic information extraction task, despite its apparent simplicity.

References

[1] S. M. Meystre and P. J. Haug, "Comparing natural language processing tools to extract problems from narrative text," in *AMIA Annu Symp Proc.*, 2005.

[2] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper and B. G. Buchanan, "A simple algorithm for identifying negated findings and diseases," *Journal of Biomedical Informatics,* vol. 34, no. 5, pp. 301-10, 2001.

[3] S. Meystre and P. J. Haug, "Natural language processing to extract medical problems," *Journal of Biomedical Informatics,* vol. 39, no. 6, pp. 589-99, 2006.

[4] O. Ghiasvand and R. J. Kate, "UWM: Disorder mention extraction from clinical text using CRFs and normalization using learned edit distance patterns," in *Proceedings of the Eight International Workshop on Semantic Evaluation (SemEval-2014)*, Dublin, Ireland, August, 2014.

[5] Q. T. Zeng, S. Goryachev, S. Weiss, M. Sordo, S. N. Murphy and R. Lazarus, "Extracting principal diagnosis, co-morbidity and smoking status for asthma research: evaluation of a natural language processing system.," *BMC Medical Informatics and Decision Making.,* vol. 6, p. 30, 2006.

[6] B. K. Bellows, J. LaFleur, A. W. C. Kamauu, T. Ginter, T. B. Forbush, S. Agbor, D. Supina, P. Hodgkins and S. L. DuVall, "Automated identification of patients with a diagnosis of binge eating disorder from narrative electronic health records.," *Journal of the American Medical Informatics Association,* vol. 21, pp. 163-8, 2014.

[7] R. H. Perlis, D. V. Iosifescu, V. M. Castro, S. N. Murphy, V. S. Gainer, J. Minnier and et al, "Using electronic medical records to enable large-scale studies in psychiatry: treatment resistant depression as a model.," *Psychological Medicine,* vol. 42, no. 1, pp. 41-50, 2012.

[8] Q. He, "Text mining and IRT for psychiatric and psychological assessment." PhD dissertation, University of Twente, 2013.

[9] R. Xu, Y. Garten, K. S. Supekar, A. M. Garber, R. B. Altman and A. K. Das, "Extracting subject demographic information from abstracts of randomized clinical trial reports," in *Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics; Building Sustainable Health Systems.*, 2007.

[10] Y. Zhang and J. Patrick, "Extracting patient clinical profiles from case reports," *Australasian Language Technology Workshop,* vol. 61, no. 50.5, 2006.

[11] D. T. Heinze, M. L. Morsch, R. E. Scheffer, M. A. Jimmink, M. A. Jennings, W. C. Morris and A. E. W. Morsch, "LifeCode: A deployed application for automated medical coding," *AI Magazine,* vol. 22, no. 2, pp. 76-88, 2001.

[12] S. B. Johnson and C. Friedman, "Integrating data from natural language processing into a clinical information system," in *Proceedings of the AMIA Annual Fall Symposium*, 1996.

[13] C. Friedman, "Towards a comprehensive medical language processing system: methods and issues," in *Proceedings of the AMIA annual fall symposium*, 1997.

[14] J. Zavrel and W. Daelemans, "Feature-rich memory-based classification for shallow NLP and information extraction," in *Text Mining, Theoretical Aspects and Applications*, 2003, pp. 33-54.

[15] E. S. Chen, S. Manaktala, I. N. Sarkar and G. B. Melton, "A multi-site content analysis of social history information in clinical notes," in *AMIA Annual Symposium Proceedings*, 2011.

[16] M. Talebi and C. Köse, "Identifying gender, age and education level by analyzing comments on Facebook," in *Signal Processing and Communications Applications Conference (SIU), 2013 21st.*, 2013.

[17] W. W. Chapman, W. Bridewell, P. Hanbury, G. F. Cooper and B. G. Buchanan, "A simple algorithm for identifying negated findings and diseases," *Journal of Biomedical Informatics,* vol. 34, no. 5, pp. 301-310, 2001.

[18] N. K. Mishra, D. M. Cummo, J. J. Arnzen and J. Bonander, "A rule-based approach for identifying obesity and its comorbidities in medical discharge summaries," *Journal of the American Medical Informatics Association,* vol. 16, no. 4, pp. 576-9, 2009.

[19] J. C. Denny, R. A. Miller, L. R. Waitman, M. A. Arrieta and J. F. Peterson, "Identifying QT prolongation from ECG impressions using a general-purpose natural language processor," *International Journal of Medical Informatics,* vol. 78, pp. S34-S42, 2009.

[20] K. J. Mitchell, M. J. Becich, J. J. Berman, W. W. Chapman, J. Gilbertson, D. Gupta, J. Harrison, E. Legowski and R. S. Crowley, "Implementation and evaluation of a negation tagger in a pipeline-based system for information extraction from pathology reports," *Medinfo,* pp. 663-7, 2004.

[21] P. G. Mutalik, A. Deshpande and P. M. Nadkarni, "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,* vol. 8, no. 6, pp. 598-609, 2001.

[22] Y. Huang and H. J. Lowe, "A novel hybrid approach to automated negation detection in clinical radiology reports," *Journal of the American Medical Informatics Association,* vol. 14, no. 3, pp. 304-311, 2007.

[23] J. Nivre, "Dependency grammar and dependency parsing," *MSI Report,* vol. 1959, pp. 1-32, 2005.

[24] S. Sohn, S. Wu and C. G. Chute, "Dependency parser-based negation detection in clinical narratives," in *AMIA Summits on Translational Science Proceedings*, 2012.

[25] M.-C. de Marneffe and C. D. Manning, "Stanford typed dependencies manual," 2008. [Online]. Available: http://nlp. stanford. edu/software/dependencies manual.pdf. [Accessed September 2014].

[26] R. McDonald, F. Pereira, K. Ribarov and J. Hajič, "Non-projective dependency parsing using spanning tree algorithms," in *HLT '05 Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, 2005.

[27] M.-C. de Marneffe, B. MacCartney and C. D. Manning, "Generating typed dependency parses from phrase structure parses," in *Proceedings of the IEEE / ACL 2006 Workshop on Spoken Language Technology*, 2006.

[28] N. Lewis, D. Gruhl and H. Yang, "Extracting family history diagnosis from clinical texts," in *BICoB*, 2011, pp. 128-133.

[29] N. Lewis, D. Gruhl and H. Yang, "Dependency parsing for extracting family history," in *HISB '11 Proceedings of the 2011 IEEE First International Conference on Healthcare Informatics, Imaging and Systems Biology*, Washington, DC, 2011.

[30] K. Fundel, R. Küffner and R. Zimmer, "RelEx—Relation extraction using dependency parse trees," *Bioinformatics,* vol. 23, no. 3, pp. 365-371, 2007.

[31] S. Pyysalo, A. Airola, J. Heimonen, J. Björne, F. Ginter and T. Salakloski, "Comparative analysis of five protein-protein interaction corpora," *BMC Bioinformatics,* vol. 9, p. S6, 2008.

[32] H. Kilicoglu and S. Bergler, "Syntactic dependency based heuristics for biological event extraction," in *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task.*, 2009.

[33] American Psychiatric Association, Diagnostic and statistical manual of mental disorders, Washington D.C.: Author, 2000.

Appendix: Annotation Guidelines

NOTE: All numeric codes have been redacted in this document in order to maintain the security of currently ongoing data collection procedures.

Gender

Text Annotation
Annotate all instances where gender is noted in the header information as "SEX: " followed by either "M" or "F". It is often found in two places, at the beginning of the document and at the end of the document where the headers are located. They should be annotated in both of these places.

Coding
If "M" is indicated, code as "Male". If "F" is indicated, code as "Female".

Age

Text Annotation
In the CHIEF COMPLAINT, HISTORY OF PRESENT ILLNESS, IDENTIFYING INFORMATION and IMPRESSION sections, annotate any instances where the patient's age is noted in numerical form or spelled out using the phrase "-year(s)-old".

Start the annotation with the noun phrase referring to the patient. If the sentence does not specifically refer to the patient but the age is still being attributed to the patient, start with the beginning of the verb phrase attributing the age to the patient. End the annotation after the phrase "-year(s)-old".

Example: "This patient is a 22-year-old female."

Coding

Use the age indicated as the code.

Admission Diagnosis

Text Annotation

In the MULTIAXIAL DIAGNOSIS section, the diagnoses to be annotated are found following "Axis I". Only the Axis I diagnoses are to be annotated. Do not annotate diagnoses that are inconclusive or are not a current problem; such cases may have one of the following words/phrases: "history", "rule out", "possible", "provisional", "probable", "remission". If a diagnosis is in "partial remission" it is still part of the current diagnoses and should be annotated.

If a current diagnosis contains a related specifier, it should be included as part of the annotation.

Example: "Major depressive disorder, recurrent severe with suicidal ideation" - the entire phrase should be annotated.

If a rule out is part of a specifier for an included diagnosis the ruled-out portion should not be included in the annotation.

Example: "Depression not otherwise specified, rule out major depressive disorder" - only select the "Depression not otherwise specified" part.

Diagnoses will usually be preceded by a number contained in parentheses, such as "(1)". These can be used to distinguish between separate diagnoses. If this notation is not used in a particular assessment, each diagnosis often ends with a period, which can also be used to help distinguish between diagnoses.

The annotated diagnosis will start with the first letter of the first word and end with the last letter of the last word. Do not annotate the period after the diagnosis.

Do not annotate any text in Axes II through V.

Coding

Assign the code to a particular diagnosis based on Table 1.

Any diagnoses indicated as "Not Otherwise Specified" (NOS) should be coded to the corresponding NOS diagnosis. For example, "Anxiety NOS with obsessive-compulsive features" would only be coded as "Anxiety NOS" without the additional code for "obsessive-compulsive disorder".

If a "depression" diagnosis is specifically noted as "major depression" or "major depressive disorder" it should have code (REDACTED), otherwise code as "Mood Disorder NOS". Simply put, if a diagnosis involves "depression" but does not have the word "major" as well, code as "Mood Disorder NOS".

If a diagnosis does not appear to fit within any of the designated diagnosis codes, code it as "Other".

If the Anorexia Nervosa diagnosis does not have the subtype specifier, code as "Anorexia Nervosa Unspecified".

Marital Status

Text Annotation

In the SOCIAL HISTORY section, annotate any individual phrases or sentences that conclusively (based on information only within this section) indicate the patient's current marital status. Imagine that you have to build a "case" to prove the patient's current marital status, and therefore must annotate any and all possible text in this section as "evidence". The assessment is not guaranteed to provide all sufficient information or be entirely accurate; despite this one should only classify the patient's marital status as it is expressed in the assessment.

Each phrase must first be considered independently of any other sentence when considering eligibility for annotation, but if multiple phrases are required to provide proper context, all relevant phrases should be annotated in addition to any other mentions of current marital status.

Do not annotate phrases that describe the patient's emotional status towards relationships ("The patient has problems with intimacy" or "The patient has poor social skills and has trouble relating to others") as this is not conclusive enough to describe the nature of the patient's actual relationship with a significant other. If there is no conclusive information or you are not sure, do not make any assumptions about the patient's marital status; that particular assessment will not have an annotated marital status.

Start the annotation with the noun phrase referring to the patient and end with the minimal amount of text needed to understand the patient's marital status. Unrelated information may be included if it is between the mention of the subject and the marital status.
Example: "The patient has been married for 20 years and has been living in Houston."

If the sentence does not specifically refer to the patient but it can be inferred that the patient's marital status is being described, start with the beginning of the first relevant verb phrase and end with the minimal amount of text needed to understand the patient's marital status.
Example: "No contact with family, and is not currently dating."

If a verb phrase cannot be found then start with the first word in the phrase and end with the minimal amount of text needed to understand the patient's marital status.
Example: "Currently divorced with severe social anxiety."

Do not include the period at the end of a sentence; if multiple sentences are to be selected, annotate them a separate individual phrases.

Coding

See Table 2 for marital status codes.

There are several potential ways that each status can be conveyed:

Single (not married)

Patients that are not married are coded as "single", although they may be dating at the time of admission.

Examples:

- The patient has no close relationships.
- The patient is not interested in dating.
- The patient has a boyfriend/girlfriend.
- The patient broke up with a boyfriend/girlfriend.
- The patient is single.

Married

The assessment will often be clear about if a patient is married, but if it only notes that the patient has been living with their husband/wife at the time of admission, that is also sufficient evidence to consider the patient married. If the patient is separated from their spouse but not divorced, consider the patient as still married. Annotate any phrases or sentences that mentions that they have a significant other that also sufficiently describes their current status as married (For example, "Her husband works as a salesman.")

Divorced

The assessment will often be clear if the patient is divorced.

Widowed

Annotate as "widowed" if it is clear that the patient's spouse has passed away and the patient has not remarried. If they are remarried, classify them as "married" since that is a more current marital status.

Education Level

Text Annotation

In the SOCIAL HISTORY section, annotate any individual phrases or sentences that conclusively (based on information only within this section) indicate the patient's current level of education. Imagine that you have to build a "case" to prove the patient's current level of education, and therefore must annotate any and all possible text in this section as "evidence". The assessment is not guaranteed to provide all sufficient information or be entirely accurate; despite this one should only classify the patient's level of education as it is expressed in the assessment.

Each phrase must first be considered independently of any other sentence when considering eligibility for annotation, but if multiple phrases are required to provide proper context, all relevant phrases should be annotated in addition to any other mentions of current level of education.

If there is no conclusive information or you are not sure, do not make any assumptions about the patient's level of education; that particular assessment will not have an annotated/coded level of education.

Start the annotation with the noun phrase referring to the patient and end with the minimal amount of text needed to understand the patient's education level. Unrelated information may be included if it is between the mention of the subject and the level of education.
Example: "She moved to Chicago after graduating from college with a degree in Chemistry."

If the sentence does not specifically refer to the patient but it can be inferred that the patient's level of education is being described, start with the beginning of the first relevant verb phrase and end with the minimal amount of text needed to understand the patient's education level.
Example: "Asked to leave college due to drug use."

If a verb phrase cannot be found then start with the first word in the phrase and end with the minimal amount of text needed to understand the patient's education level.
Example: "High school graduate."

Do not include the period at the end of a sentence; if multiple sentences are selected, annotate them as separate individual phrases.

Coding

See Table 3 for education codes.

If the patient has been accepted or enrolled in a college but has not attended yet, code as "High School Graduate"

If it notes that a patient has attended a college but there is no indication of receiving a degree, code as "Some College". This code is used regardless of whether the patient pursuing a 4-year degree or an associate degree. Similarly, if a patient has attended any kind of a higher level of education beyond a college (4-year or associate) degree but has not finished, code as "Some Graduate School".

Table 1: Diagnosis Codes

OCD
Trichotillomania
Panic Disorder
PTSD
Learning Disability
Substance Abuse/Dependence
Social Anxiety
Body Dysmorphic Disorder
Generalized Anxiety Disorder
Anorexia Nervosa Restricting Subtype
Anorexia Nervosa Binge/Purge Subtype
Bulimia Nervosa
Eating Disorder Not Otherwise Specified (EDNOS)
Major Depressive Disorder
Dysthymia

Bipolar Type 1
Bipolar Type 2
Mood Disorder Not Otherwise Specified (Mood Disorder NOS)
Aspergers Spectrum
Autism
Oppositional Defiance
ADHD
Other
Tic/Tourette’s
Presence of a Personality Disorder
Anxiety Not Otherwise Specified (Anxiety NOS)
Bipolar Not Otherwise Specified (Bipolar NOS)
Anorexia Nervosa Unspecified

Table 2: Marital Status Codes

Single
Married
Divorced
Widowed

Table 3: Education Codes

Some High School
High School Graduate
Some College
Associate Degree Graduate
College (4-Year) Graduate
Some Graduate School
Graduate School Graduate (MS, MA)
Graduate School Graduate (PhD, JD, MD)