IDENTIFYING OUTCOMES OF CARE FROM MEDICAL RECORDS TO IMPROVE DOCTOR-PATIENT COMMUNICATION

A Thesis

by

SETH C. POLSLEY

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Tracy Hammond
Co-Chair of Committee,	Yoonsuck Choe
Committee Member,	Marcia Ory
Head of Department,	Dilma Da Silva

August 2017

Major Subject: Computer Engineering

ABSTRACT

Between appointments, healthcare providers have limited interaction with their patients, but patients have similar patterns of care. Medications have common side effects; injuries have an expected healing time; and so on. By modeling patient interventions with outcomes, healthcare systems can equip providers with better feedback. In this work, we present a pipeline for analyzing medical records according to an ontology directed at allowing closed-loop feedback between medical encounters. Working with medical data from multiple domains, we use a combination of data processing, machine learning, and clinical expertise to extract knowledge from patient records. While our current focus is on technique, the ultimate goal of this research is to inform development of a system using these models to provide knowledge-driven clinical decision-making.

ACKNOWLEDGEMENTS

The vast majority of this work was performed under the supervision of and with assistance from Dr. Duane Steward of the Texas A&M Health Science Center (HSC) and his College of Medicine Biomedical Informatics Research (COMBIR) group. Dr. Steward and the other members of the COMBIR group, in particular Atif Tahir, Dr. Muppala Raju, and Dr. Akintayo Akinleye, were instrumental in the development of these ideas, and the author owes them an immense debt of gratitude.

The author would also like to thank each of the pilot investigators for the time they have given providing insight and reviewing our preliminary works: Duane Steward, DVM, MSIE, PhD; Craig Borchardt, PhD; James N. Burdine, DrPH; Nancy Dickey, MD; Sharon Kerwin, DVM, MS; Allen J. Roussel, DVM, MS; Brett Trusko, PhD, MBA; and Stephen Wong, PhD, PE.

Finally, Dr. Tracy Hammond, the members of the Sketch Recognition Lab (SRL), Dr. Ruihong Huang, and the members of the TAMU Natural Language Processing Group have been very supportive collaborators whose assistance has encouraged progress multiple times.

CONTRIBUTORS AND FUNDING SOURCES

This work was supported by a thesis committee consisting of Professor Tracy Hammond (advisor) of the Department of Computer Science and Engineering, Professor Yoonsuck Choe of the Department of Computer Science and Engineering, and Associate Dean of Research/Regents and Distinguished Professor Marcia Ory of the Department of Health Promotion and Community Health Sciences.

Extensive support was also provided by Assistant Research Professor Duane Steward of the Department of Biomedical Informatics in the College of Medicine at HSC, along with the students in his lab, COMBIR. Drs. Muppala Raju and Akintayo Akinleye provided their medical expertise when reviewing data and are referenced frequently in this work as our expert annotators. Atif Tahir, Girish Kasiviswanthan, and Pooja Jhunjhunwala were instrumental in developing the pre-processing pipeline and provided assistance in building the early processing algorithms. Stuart Slattery, Preston Baker, and Amber Schulze helped collect the initial data and perform analysis. Assistant Professor Ruihong Huang of the Department of Computer Science and Engineering offered many excellent suggestions regarding our natural language processing techniques. All other work conducted for the thesis was completed by the student independently.

Two sources contributed data that made this research work possible. The hospice data set was provided by Hospice Brazos Valley (HBV), and the veterinary data set was supplied by the Texas A&M Veterinary Medical Teaching Hospital (VMTH).

Primary funding for this work was provided by the HSC Clinical Science & Translational Research Institute (CSTR or CST*R) Pilot Grant #CSTR2016-2, "Determination of the Competency of Electronic Health Record Data to Support Outcomes

Feedback for Medical Interventions.” This grant was awarded to support a pilot investigation as part of a larger project to “engineer a new knowledge-based system of per encounter outcomes feedback... that empower[s] outcomes feedback with adequate potency to impact effective delivery of care.”

NOMENCLATURE

CAD	Computer-Aided Diagnosis
CLAMP	Clinical Language Annotating, Modeling, and Processing toolkit
COMBIR	College of Medicine Biomedical Informatics Research
CRF	Conditional Random Field
CSTR or CST*R	HSC Clinical Science & Translational Research Institute
EMR	Electronic Medical Record System
FN	False Negative
FP	False Positive
HBV	Hospice Brazos Valley
HCD	Hospice Care Data
HIPAA	U.S. Health Insurance Portability and Accountability Act
HMM	Hidden Markov Model
HSC	Texas A&M Health Science Center
IHTSDO	International Health Terminology Standards Development Org
IOO	Integrated Outcomes Ontology
IR	Information Retrieval
KNN	K-Nearest Neighbor
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NN	Neural Network
PCA	Principal Components Analysis
POS	Part of speech
PHI	Protected Health Information

SNOMED	Systemized Nomenclature of Medicine
SRL	Sketch Recognition Lab
SVD	Singular Value Decomposition
TAMU	Texas A&M University
TN	True Negative
TP	True Positive
VCD	Veterinary Care Data
VMDB	Veterinary Medical Database
VMIS	VMTH Medical Information System
VMTH	Texas A&M Veterinary Medical Teaching Hospital

TABLE OF CONTENTS

Page

ABSTRACT . . . ii

ACKNOWLEDGEMENTS . . . iii

CONTRIBUTORS AND FUNDING SOURCES . . . iv

NOMENCLATURE . . . vi

TABLE OF CONTENTS . . . viii

LIST OF FIGURES . . . xi

LIST OF TABLES . . . xiii

1. INTRODUCTION . . . 1

   1.1 Overview . . . 1
   1.2 Research Questions . . . 2
   1.3 Outline . . . 3

2. BACKGROUND . . . 4

   2.1 Ontologies and Biomedical Informatics . . . 4
   2.2 Redaction of Medical Records . . . 5
   2.3 Medical Records Analytics . . . 6

3. INTEGRATED OUTCOMES ONTOLOGY . . . 9

   3.1 Ontology Competency Questions . . . 9
   3.2 Roles . . . 11
      3.2.1 Patient . . . 11
      3.2.2 Observing Caregiver . . . 11
      3.2.3 Prescribing Provider . . . 12
   3.3 Encounters . . . 12
   3.4 Interventions . . . 13
   3.5 Outcomes . . . 14

4. PRE-PROCESSING 16

4.1 Data Sources 16
4.1.1 Veterinary Care 17
4.1.2 Hospice Care 17
4.2 Ontology-Driven Redaction 18
4.2.1 Structured 19
4.2.2 Unstructured 21
4.2.3 Complete Pipeline 22
4.3 Validation 22
4.3.1 VCD 23
4.3.2 HCD 25

5. ANALYSIS OF THE STRUCTURED RECORD 28

5.1 Features of Encounters 28
5.1.1 Diagnosis Codes 29
5.1.2 Presenting Complaint 32
5.1.3 Medications 33
5.1.4 Age 33
5.2 Episode Identification 33
5.2.1 Baseline Using Age 35
5.2.2 Combining Features 36
5.3 Interventions and Outcomes 37

6. ANALYSIS OF THE UNSTRUCTURED RECORD 39

6.1 Segmenting Discharge Summaries 39
6.2 Word Embeddings 43
6.2.1 Word2Vec 45
6.2.2 Brown Clusters 48
6.3 Expert Annotation 49
6.4 Computer Annotation 51
6.4.1 Features 51
6.4.2 Evaluation 56

7. FUTURE WORK 61

8. CONCLUSION 62

8.1 Research and Competency Questions Revisited 62
8.2 Final Remarks 64

REFERENCES 65

APPENDIX A. DATA PROCESSING AND REDACTION MAPPINGS . . . 81

APPENDIX B. WORD CLOUDS FOR DISCHARGE SUMMARY SEGMENTS 86

LIST OF FIGURES

FIGURE Page

3.1 The feedback-based ontology showing support for outcomes of care, including roles and interventions. 13

5.1 Principal Components Analysis of diagnosis codes. 32

5.2 A scatter plot showing multiple encounters relative to each other in a vector space comprised of age, medications, and diagnosis codes. . . . 34

5.3 Area chart comparing the baseline model against the multidimensional one for hierarchical clustering of encounters into episodes. 36

6.1 Different classifier performance on labeling three classes based on bag-of-words. 41

6.2 Bag-of-words scoring vectors decomposed to a three dimensional representation. 42

6.3 Stylized word cloud showing weighted significance of the most frequent words in all the VCD documents. 44

6.4 Word2Vec embeddings in two dimensions using PCA for the most frequent 100 words in VCD; we see that some semantics are retained through word proximity even at this crude level. 45

6.5 Average sum of squares distance within the clusters for different values of k when building K-Means clusters to reference the expansive Word2Vec model. 47

6.6 The overall F1-score for the CRF on VCD interventions according to different window sizes and feature sets. Window sizes vary from the single word to the past three words and next two. Feature sets consider only topology, topology + part of speech, and topology + part of speech + semantic clustering. 52

6.7 Individual class F1-scores in the -3:+2 word window. ‘Word’ uses only the word stem, while ‘topology’ includes other aspects like the suffix. ‘PartOfSpeech’ remains the same as before, but ‘Brown’ and ‘Word2Vec’ have been split into two stages from the former ‘Semantics’ label. 53

6.8 Radar chart showing F1-scores by class according to the feature set. . 54

B.1 Word cloud visualizing the top 60 words in the “treatment” subsection. 87

B.2 Word cloud visualizing the top 60 words in the “exercise” subsection. 87

B.3 Word cloud visualizing the top 60 words in the “diet” subsection. . . 88

LIST OF TABLES

TABLE Page

4.1 Sample dictionary of names. 19

4.2 Error rates by document type in the first iteration of VCD redaction. 24

4.3 Performance metrics by document type in the first iteration of VCD redaction. 24

4.4 Error rates by document type in the second iteration of VCD redaction. 25

4.5 Performance metrics by document type in the second iteration of VCD redaction. 25

4.6 Error rates for the HCD redaction considering both all names and the pure HIPAA-defined PHI. 26

4.7 Performance rates for the HCD redaction for all names, including clinicians, and true PHI. 26

5.1 Sample VMDB diagnosis codes showing similarity among different code groups. 30

5.2 Sample VMDB code conversion at the digit-level. 31

6.1 Several segments of discharge summaries and their 10 most frequent words. 42

6.2 Five nearest words to the 10 cluster centers selected by K-Means in VCD. 48

6.3 Sample Brown clusters. 49

6.4 Kappa scores denoting inter-rater agreement for the intervention and outcome corpus. 51

6.5 Features of each word in the CRF word window; the number is the distance from the current word (0). Key: Brown - Brown cluster, POS - POS tag, IsCap - Starts with a capital letter, IsUpper - Word is uppercase, Stem - Porter stem of the word, Suffix - Last 3 characters, W2V - Cluster number in the Word2Vec vector space 55

6.6 Performance metrics for VCD CRF trained to extract intervention/ therapy labels 56

6.7 Confusion matrix for the VCD Intervention CRF based on human validation. 57

6.8 Performance metrics for VCD CRF trained for outcomes-oriented labels. 58

6.9 Performance metrics for HCD CRF trained to extract both intervention and outcome labels. 59

A.1 VCD mappings from the original VMIS data. 82

A.2 HCD mappings from the original HBV data. 83

A.3 HCD mappings for supplemental table data from HBV. 84

A.4 HCD support information referenced from HBV tables. 85

1. INTRODUCTION

1.1 Overview

Hand-written medical patient records have been almost completely replaced by digital storage. Electronic medical record systems (EMRs) offer many advantages over paper files in terms of size, speed, management, and aggregated reports [1]. When available, large medical digital data sets have provided a wealth of machine learning data for researchers to use in recent years, and great advancements have been made in areas like gene sequencing and computer-aided diagnosis (CAD) systems [2, 3]. These tools can certainly help doctors provide better care, but the application of advanced modeling and data science algorithms remains largely unexplored regarding the practice of medicine in outpatients. In such cases, a patient is discharged with instructions and suggestions for follow-up visits, but left on their own to report outcomes [4, 5]. As a result, outcomes of care are rarely communicated [6], and even if they are, they may be narrowly focused and limited in scope [7, 8]. Generally, results are reconstructed during subsequent encounters, which can leave health care delivery under-informed. Capturing outcomes of care systematically could provide better information and save time during follow-up visits, where studies have shown that nearly 50% of the time physicians spend in the office is spent on electronic record keeping and paperwork [9, 10].

Patient records are comprised of structured and unstructured text components, a growing volume of knowledge reflected in the diversity of patient care. However, utility is limited by the constraints of rigid standards for claims reimbursement, inconsistent documenting practices, and federal law. In the United States, the Health Insurance Portability and Accountability Act (HIPAA) limits the ability of health care providers to spread protected health information (PHI), and the cost of violations can be very high, including hundreds of thousands of dollars and prison time [11]. As a result, any medical data used for research must be rigorously reviewed for release, and availability often remains restricted. Another barrier is the preponderance of free text in medical records, driving a need to support identifying structure in clinical text which has long been recognized as a prerequisite to algorithmic analysis [12]. Most machine learning techniques also require labeled data for training, so the lack of annotated clinical text for natural language processing (NLP) remains another restraint [13].

1.2 Research Questions

In this work, we will focus on the development of a cross-domain proof-of-concept methodology for organizing and processing electronic medical records so that outcomes-oriented data can be automatically extracted. In part, our motivation is to help reduce the time spent managing records and reconstructing outcomes of former encounters. We also wish to alleviate some of the existing limitations in research involved medical data with the development of an ontology-driven redaction engine to support end-to-end processing of patient records and an annotated corpus for machine learning tasks related to interventions and outcomes of care in clinical text. Together, these concerns form the basis of a pipeline with the potential to revolutionize care plan management through better reporting of outcomes. Identifying the symptoms of surgical site infections or recognizing the need to discontinue a certain medication due to side effects are just two simple examples illustrating the value of closed-loop outcomes-oriented feedback.

This work is centered around answering the following research questions:

1. Can cross-domain medical records be encoded in a single, feedback-based ontology representing outcomes-oriented concerns?
2. What features are necessary to divide patient records into related segments of episodes of care?
3. Can we extract interventions and, in particular, outcomes data from free medical texts in the absence of a structured record?

1.3 Outline

The remaining content is outlined sequentially, matching the order of exploratory efforts undertaken to answer each of these questions. First, in Chapter 2, is a common background providing an overview of related and prior work to each task in the pipeline: ontology development, pre-processing and redacting of medical data, structured record analysis, and clinical text annotation and mining. Then, our methodologies, including the most relevant literature, and evaluations are provided in greater detail for each task. Chapter 3 discusses the necessary, underlying ontology which is used to inform all later outcomes-oriented development, alongside the competency questions it seeks to answer. Chapter 4 describes the pre-processing our data underwent, ontology-driven redaction, and the clinical validation of the cleaned data. In Chapter 5, a set of analyses of the structured portion of the data is assessed. The unstructured text, which holds some of the richest source of outcomes-oriented information, is the subject of a number of studies, annotations, and evaluations in Chapter 6. Ongoing work, future avenues of research, and ideas for potential applications are reviewed in Chapter 7. Finally, we close in Chapter 8 by revisiting the research questions in the context of our findings before moving on to an overview and general conclusion. Appendices and references comprise the final pages.

2. BACKGROUND

Some motivating works for analyzing outcomes of care with more computational vigor have been shared in the previous section. In general, the literature recognizes the current methods of recognizing outcomes are limited; they are either rarely captured from the patient or not well-documented in the EMR [5, 4, 8, 14, 7]. We turn to ontologies, which are models defined by formal terms and relationships, to assist in the task of better documenting outcomes in existing records. An ontology allows cleaning the data in such a way that supports algorithmic analysis. In this section, we provide background on each of the relevant tasks.

2.1 Ontologies and Biomedical Informatics

One of the more fundamental aspects of recent biomedical informatics research has been a focus on ontology-driven development [15, 16, 17, 18, 19, 20, 21]. In a domain like healthcare – where information is dense, diverse, and specialized – an ontology allows representing knowledge in a usable manner, because it describes a framework for clearly defining known terms and their relationships [22, 23, 24, 25, 26]. Once the data has been formally described via an ontology, new applications become apparent. To provide several examples, simply by formalizing electronic records as an ontology, researchers have shared better ways to represent patient care profiles [27], perform risk assessment [28], evaluate elderly care [29], and more [30, 31].

The greatest promise lies in ontology-driven computational models, where the structure of an ontology makes the data accessible to programmatic operations. Perhaps the best known example is computer-aided diagnosis (CAD), in which a computer analyzes patient data to generate a patient diagnosis [32, 33, 34]. Biomedical informatic researchers have explored other applications as well, such as identifying groups of similar patients [35], mining for co-morbidities [36], enabling improved management of Parkinson's disease [37], and automatically assisting with patient care coordination based on free text [38], to name just a few specific examples.

2.2 Redaction of Medical Records

Knowledge buried in medical text is valuable, but protection of Protected Health Information (PHI) is a special concern when dealing with medical records data as it is protected by law [19]. Appropriately, many efforts have been made to build reliable de-identification pipelines. Most existing methods rely on rule-based systems that match patterns and dictionaries of expressions that frequently contain PHI. Sweeny built one such tool called Scrub, which uses templates and a context window to replace PHI [39]. Datafly, another program developed by Sweeny, offers user-specific profiles, including a list of preferred fields to be scrubbed [40]. Thomas developed a method that uses a lexicon of 1.8 million names to identify people along with "Clinical and Common Usage" words from the Unified Medical Language System (UMLS) [41]. Miller developed a de-identification system for cleaning proper names from records of indexed surgical pathology reports at the Johns Hopkins Hospital [42]. Proper names were identified from available lists of persons, places and institutions, or by their proximity to keywords, such as "Dr." or "hospital." The Perl tool *Deid* is a recent development which combines several of these rule-based and lexical approaches with some additional capabilities like better handling of time [43]. Rule-based approaches have been widely used with good success, but there are alternatives. For instance, Dernoncourt applied recurrent neural networks to the task of identifying PHI to remove the need for large dictionaries [44]. South introduced "pre-annotation" to improve retention of some information content, breaking down all PHI by type in detail, a necessary aspect of redacting while retaining value [45].

2.3 Medical Records Analytics

In [46], the authors suggest some approaches researchers can use to help mitigate the lack of medical data available, among them being the suggestion to release clearly-annotated data, but as described in [47], annotation can be time-consuming, difficult, and narrowly-targeted. To provide just a few examples, consider some existing corpora: [48] focused on identifying terms related to ovarian cancer; [49] looks specifically at medications; [50] developed a data set for improving part-of-speech tagging on medical text; [51] looks at temporal relationships; and [52] use annotation to improve co-reference resolution. These works and their respective data sets have been undeniably influential and beneficial to the research community, but there continue to be gaps in supporting generalizable solutions to multiple domains. Some works have addressed generalization by providing more comprehensive sets of patient records [53] or with a prescriptive framework for annotation [54, 55, 56, 57]. Even with experts and annotation software, there is still the difficulty of creating generalizable annotation schema. The authors of [57] and [55] explain well their annotation schema, but the approach used in [58] is particularly memorable because it utilizes an existing ontology, as we too rely on an ontology for structure.

Of particular interest to this work is text processing which seeks to extract relevant information from free medical texts like clinical notes and discharge summaries. As with the structured record, extracting this information following an ontology presents many new applications, but there is a prerequisite challenge of recognizing the data in the first place since it is not necessarily elsewhere in the patient file [13, 47]. Many biomedical informatics researchers have tackled parts of this problem, be it in the form of algorithms, toolkits, or ideas. In [33], the authors use random forest classifiers to weight sentences from clinical notes as relevant to diagnosis or not. The authors of [34] use bag-of-words features to assist the prediction of presence of cancer. More related to our proposed approach, [38] develops an ontology to define care-related terms and extracts types of care “activities” from clinical notes using regular expressions. The Clinical Language Annotating, Modeling, and Processing Toolkit (CLAMP) has also been used extensively on clinical texts and discharge summaries [59]. CLAMP’s pipelines are generally built around conditional random fields (CRFs) or Neural Networks (NNs) to support tasks like semantic role labeling or entity recognition [60, 61, 62, 63].

CRFs, which are similar to Hidden Markov Models (HMMs), and NNs are a common approach to automatic text annotation [64, 65, 66, 67]. A CRF essentially describes a set of states and learns the likelihoods of transitioning among them [62]. These models rely on a number of practices from natural language processing (NLP) to develop their features. Prominent features are topological in nature. These are the word itself, case, part-of-speech (POS), root word, suffix, or other characteristics. Additional features include semantic labels, n-gram overlap, hyponymy, hypernymy, and other relationships.

Our efforts builds on many of these works. For instance, feature encoding of medical terms will be used to inform computational methods at later stages in our pipeline [30, 68]. Applications like prediction of clinical events or association rule mining will be tied to outcomes-oriented feedback [36, 69]. The overarching goal of detecting outcomes-oriented feedback appears very rarely in related literature. Ontology reference tools like “Ontobee” show that certain outcome terms exist in certain ontologies [70], but their use is limited. In [71], the authors discuss the value of assessing outcomes, and while they present an objective technique, there is no ontology encoding. The need to report outcomes is motivated in [72], but rather than attempting to capture information automatically, only a means of describing adverse outcomes is presented. Our strategy is to encode both positive and negative outcomes in an ontology which can bind outcomes automatically to a care encounter, informing later encounters and closing the feedback loop between encounters.

3. INTEGRATED OUTCOMES ONTOLOGY

While not the emphasis of this work, all subsequent steps build on the outcomes-oriented ontology developed to answer our research questions. An ontology is useful prior to working with data because it formalizes the desired structures and smooths the cleaning and pre-processing steps. We present the “Integrated Outcomes” ontology (IOO), so-named for its prominent feedback loop tying outcomes to subsequent encounters, according to **roles**, **encounters**, **interventions**, and **outcomes**. Most of these components are measurable parts of the medical record and form the basis of the relationships governing the ontology.

3.1 Ontology Competency Questions

A set of core competency questions was used to develop IOO:

1. Can the actors filling the following roles be identified for each encounter?

2. Can related encounters be identified, distinguished and worked with programmatically independent of unrelated encounters?

3. Can the intervention(s) of the encounter be identified, specifically those related to outcomes that determine whether care should be continued or altered?

4. Can the outcomes related to the specific interventions above be identified, distinguished and worked with programmatically independent of unrelated outcomes?

5. Among the outcomes related to the relevant interventions, can the intended outcomes be identified and differentiated from unintended outcomes? Also can

potential outcomes be identified and distinguished from actual outcomes that have occurred?

6. Can therapy change interventions be classified as one of initiate, continue, alter, or discontinue therapy?
7. Can sub-classes of therapy be identified, distinguished and worked with programmatically with regards to specifically related outcomes (e.g, medication, bandaging, preventative, exercise, diet, procedure, etc.)?

Corresponding ontological elements were selected around being able to answer one or more of the posed questions:

- Roles – #1
- Encounters – #2
- Interventions – #3, #6, and #7
- Outcomes – #4, #5, and #7

Notice that several of the questions incorporate programmatic analysis. While the programmatic portion is not covered in this chapter, later methods and results will relate back to the IOO competency questions. These competency questions associated with our earlier research questions to a certain extent as well, which will become more apparent throughout our methodology.

Before discussing each aspect in greater detail, we should note that IOO is not intended to capture all the information in a medical record. Many existing ontologies have been shared which attempt to formalize as much data from as many sources as possible, as in [22] or [73]. IOO is designed to be as simple as possible to support identifying outcomes, but this means not all information is saved. What is saved may be reduced in complexity for the purposes of IOO's modeling. This was a deliberate choice made to promote generality and scalability over a complex ontology that can store all necessary information standing alone. In other words, IOO is intended enable outcomes-oriented decision making in multiple medical domains, not describe the structure for general EMR data.

3.2 Roles

For our purposes, a “role” is a living entity who fills a position relative to another role and affects or is affected by other entities. There are many roles in medical data – doctors, receptionists, patients, insurance claims processors, family, therapists, and many more. In [31], the authors compile a list of 220 roles from seven ontologies, finding that any single one is inadequate at covering all roles. As stated, IOO's simple design precludes the need to represent all 220 roles. For codifying outcomes of care, we define only three distinct roles.

3.2.1 Patient

The first and most obvious role is the patient. A patient is critical because he or she is the one who receives medical intervention and experiences outcomes. For each record, a patient is a unique entity; a different patient would be the focus of a different record. Patients are the only unique role.

3.2.2 Observing Caregiver

The observing caregiver may also be called the “outcome observer.” This is the person recognizing an outcome has occurred, either desirable or not. In certain domains, this will be the patient. In others, e.g. veterinary care or pediatrics, there will likely be someone else acting as the observing caregiver. The caregiver is vital in regard to outcomes because he or she reports them. The interval between encounters may also be heavily influenced by the observing caregiver because the time taken to observe the outcome and the time taken for the outcome itself will differ. When there is only the patient, the outcome may be observed more immediately. There may be multiple observing caregivers for a single patient.

3.2.3 Prescribing Provider

The provider is the medical entity prescribing an intervention which will lead to an outcome. This can be a doctor, nurse, therapist, or some other professional who has interacted with the patient and/or observing caregiver during an encounter. As with the observing caregiver, there may be multiple prescribing providers for a patient, and often, there will be because different doctors or clinics may treat the same patient depending on the current condition.

3.3 Encounters

An "encounter" is an interaction between a patient and/or observing caregiver and a prescribing provider. Usually, this will be an in-person visit at a clinic, although encounters may take other forms of client communication. Because the encounter describes the interaction between several roles, it contains the medical interventions that will lead to outcomes.

Sequences of encounters also have an interesting relation since they may be tied together with manifestations of an outcome. "Episodes" of encounters, which are sets of encounters grouped by a similar problem and, therefore, similar interventions, will internally be linked by outcomes. Episodes usually change for different problems because the expected outcomes will likely also change. Thus, between episodes, encounters are not typically linked by outcomes.

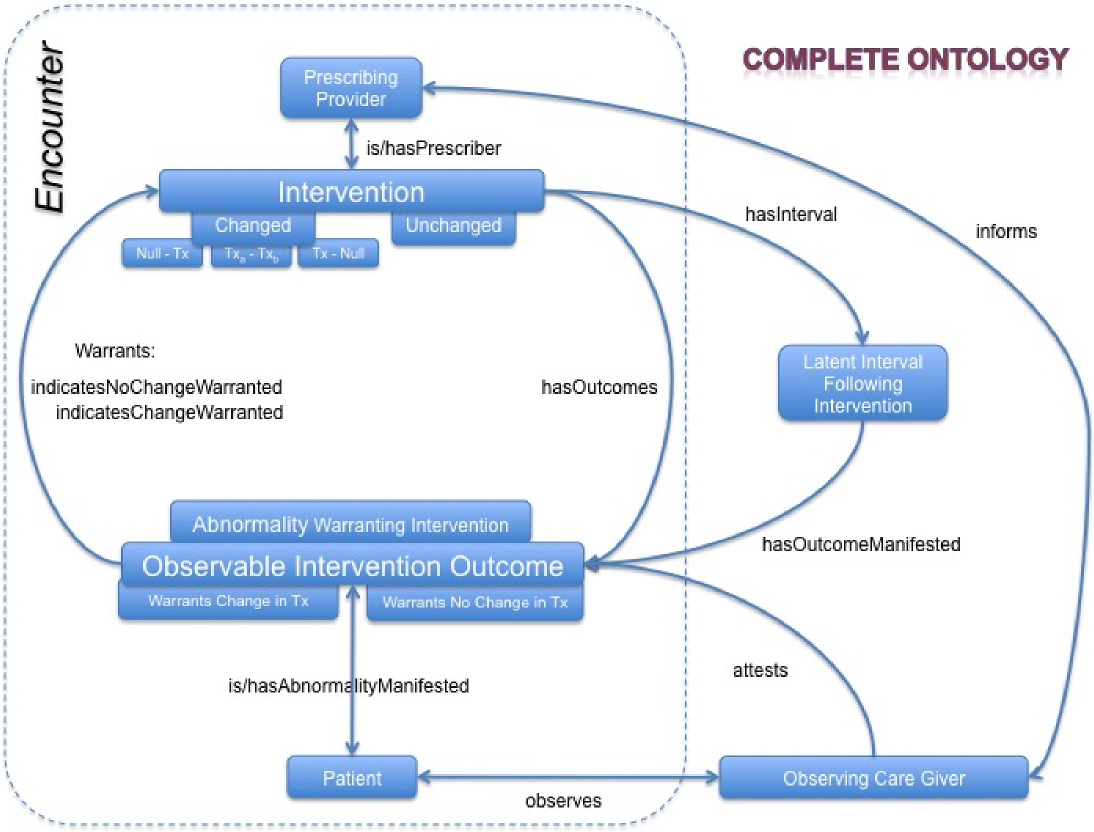

Figure 3.1: The feedback-based ontology showing support for outcomes of care, including roles and interventions.

3.4 Interventions

The "intervention" is some form of medical prescription given by the prescribing provider. Most often, this will be medicine, but it can also be a therapy, dietary change, or some other restriction. Interventions have the greatest potential to be affected by capturing outcomes in a feedback loop because they are most-closely linked to outcomes. Interventions should both be informed by the previous outcome and lead to the next.

For the purposes of IOO, interventions are contained within encounters, which are linked by outcomes. Thus, we can describe the effect of the outcome on the next intervention by breaking down interventions into four classes:

- *Initiate Therapy* - Changed from no previous to a new intervention
- *Continue Therapy* - Unchanged between encounters
- *Alter Therapy* - Changed from a previous intervention to a new, different intervention
- *Discontinue Therapy* - Changed from a previous intervention to no intervention

Each class is straightforward except *Alter Therapy.* Several instances of altering a therapy exist, ranging from changing only the dosage or specific type of therapy all the way to prescribing a completely different medicine.

3.5 Outcomes

The final important component of IOO is the "outcome" itself. An outcome is the result of an intervention; it may be desirable or undesirable. We consider an outcome to be observable since generally the intervention is effective and the outcome is probably desirable or a complication occurs and the outcome is not. In either case, the change or lack thereof would be observed.

Outcomes are associated with an interval of time from the intervention. These are typically difficult to locate in EMRs because they are pieced together during the next encounter rather than saved to the system in real time. Similarly, outcomes may not be stored in any structured part of an EMR, appearing only in the clinical notes or presenting complaint of a subsequent visit. IOO was motivated largely by the need to identify outcomes in existing EMRs. Its formality will help support machine learning techniques that can automatically learn outcomes and their intervals, promising new ways of augmenting clinical decision making.

At this point, we have walked through all the major parts of IOO. The complete ontology is shown, including a more graphical display of the relationships, in Figure 3.1. We hope that the intervention and outcome sections have helped illustrate the value of an ontology like IOO. It is true that some existing works have considered outcomes, like the reporting system in [72] or the prescriptive framework in [71], but even when the authors have had similar goals to our work, like the ontology-driven clinical text mining system in [33], they focus on diagnoses over outcomes due to the popularity of computer-aided diagnosis today.

4. PRE-PROCESSING*

Medical health records data has immense potential for research in furthering the field of automated healthcare. Like all data, in order to be analyzed more rigorously, it must be pre-processed, a series of steps comprised of cleaning and normalizing the digital record. One of the challenges facing biomedical informatics is the dissemination and sharing of digital records for research due to the additional strict regulations regarding patient confidentiality. Protecting private health information (PHI) is a critical responsibility of health care providers, with the U.S. Health Insurance Portability and Accountability Act (HIPAA) outlining a number of principles. Removing PHI can also mean removing critical parts of a record, so building redaction techniques that preserve as much information about the original data as possible while still retaining anonymity is an important pre-processing step.

First, we will provide an overview of the data sources used throughout the remainder of this project. Next, we will turn to the pre-processing steps which are predominantly built around an ontology-driven redaction procedure built in Python.

4.1 Data Sources

IOO was designed to support multiple EMRs. Our current pipeline has been deployed on two existing data sets, one veterinary and the other hospice care. Ongoing works will feature other domains like community health providers and hospitals.

*Portions of this chapter reprinted with permission from "Role-Preserving Redaction of Medical Records to Enable Ontology-Driven Processing" by Polsley et al., 2017. Proceedings of BioNLP 2017, 194-195, Copyright 2017 by ACL [74].

4.1.1 Veterinary Care

The larger of our two data sets, the veterinary care records were provided by the Texas A&M Veterinary Medical Teaching Hospital (VMTH) in College Station, TX. The EMR, formally the VMTH Medical Information System, is called VMIS for short and was developed in-house by the university before digital records were commonly used. VMIS features files for approximately 300,000 patients, including digitized records dating back to the 1970's, over half a million appointments, and gigabytes of medical text. For the remainder of the text, the veterinary care data will be referred to as VCD for simplicity.

Despite that all of our other data regards human-centered care, we consider veterinary care an important component of this work. The *One Health Initiative*[1] is a growing idea in health fields, and one of its stated goals is to encourage collaborative research across veterinary and human disciplines. While we are not yet to explicit applications, one small example of the benefit of a *One Health* perspective is the ability to completely model zoonotic diseases [75, 76, 77, 78, 79]. These diseases spread from animals to humans, and they are gaining particular attention from mosquito-borne pandemics like Zika [80, 81, 82] but also consider diseases originating in companion animals [83]. All of our data is run through the same pipeline, so we expect to explore unique interactions between veterinary and human data at a later time.

4.1.2 Hospice Care

Our hospice care data, HCD for consistency, is provided by the Hospice Brazos Valley (HBV) clinic in Bryan, Texas. The data provided includes nearly 8,000

[1] http://www.onehealthinitiative.com

patients, each with regular in-home visits from nurses and doctors, totalling to encounters and interventions in the tens of millions. Hospice is a slightly distinct domain from other types of human treatment because the expected outcome is not necessarily obvious. In a usual case, one would think the desirable outcome is to solve the cause of the problem, but in hospice care, the problems are often terminal. Thus, expected outcomes tend to be pain management or treating symptoms. HBV's EMR stores outcomes to some extent and doesn't have nearly as much free text as the VCD, with about half a gigabyte of clinical notes.

4.2 Ontology-Driven Redaction

To manage cleaning, normalizing, and securing our data, we employ a redaction framework for removing PHI from medical records through de-identification. One of the primary goals of this framework is to preserve valuable information like roles, semantics, and time intervals as much as possible. Because this forms the preprocessing stage of future text processing, we elected to model roles according to the formal IOO; this maintains relationships and enables straightforward detection of ontological terms in later phases.

The core reasoning for our methodology is that knowing the role of a redacted name can be vital. For instance, was a condition reported by the caregiver or by the clinician? That is just a single question illustrating the potential for confusion when names are redacted without roles, yet, there is no need to blindly attempt to extract roles from free text. Nearly every EMR maintains structured data like a patient's name, family contact, and attending physician. By leveraging this knowledge, pseudonyms can be constructed that remove confusion regarding roles in the final text.

Table 4.1: Sample dictionary of names. Reprinted with permission from [74].

Patients	**Caregivers**	**Providers**
Original		
Patricia Jones	Michael Jones	Daniel Moore
	Barbara Davis	Mary Johnson
Redacted		
$Clark$	$Clark_{CAREGIVER1}$	$Clark_{PROVIDER1}$
	$Clark_{CAREGIVER2}$	$Clark_{PROVIDER2}$

The redaction pipeline operates on data in two stages to support better identification of roles in the text. First, the structured data is used to extract whatever knowledge is available, typically roles like doctors and patients, to perform knowledge-based redaction. Second, the unstructured text undergoes entity recognition to clean missed terms. While this approach requires some insight about the data beforehand, it is a logical means of ensuring we can remove all PHI without damaging roles and relationships.

4.2.1 Structured

The first round of redaction is performed using structured data in the EMR. We explain the principles of each step here, but Appendix A lists the detailed mappings between the original structured data from each source and the resulting usage or storage location.

4.2.1.1 Patient-Centric Role Preservation

Our system initially builds a dictionary of known individuals in each role. A person can have any number of names of any length but all of them are drawn directly from the fields in the EMR. In accordance with the ontology, patients will be identified first as the subject of care, a unique field in most systems. Depending on the domain, there will be a personal doctor, an attending physician, or some other clinician name given in a separate field. Caregivers may be drawn from locations like billing or family contacts. For this part, knowledge of the data structure is necessary, but once the source fields are identified, they will be consistent across the other records.

Once the dictionary of names and roles is built, patients are assigned a pseudonym randomly from a list of non-matching family names to provide anonymity and linked to the pseudonym in the dictionary. Subsequently, all individuals associated with that patient are assigned a derivative pseudonym denoting their role. Consider the example shown in Table 4.1. For this small dictionary of a single patient, we see more than one caregiver and provider listed. The system first replaces the patient's name, Patricia Jones, with a false name, Clark. This identifier then becomes the basis for all subsequent individuals with a connection to the patient.

After the dictionary has been constructed, the system knows all the original names and their new pseudonyms. The medical texts are scanned for any occurrence of any known name, ignoring case or modifiers like possessive forms. Full names will be on file, but given names and family names may appear separately in the record. Regular expressions are used to match variants of names while enforcing order.

4.2.1.2 Date Offsets

It is worth emphasizing the importance of dates in medical record data. One can simply remove or replace dates to redact PHI, as with names, but just like names, we wished to preserve more information in support of the ontology. In particular, intervals between encounters or patient ages under 89 are compliant with HIPAA and useful for tasks like association mining. A common solution is to use offsets for dates because the original date will be erased from the document without losing intervals. However, an unconstrained random offset still loses information. For instance, intervals given in the free text will be broken if a day of the week is mentioned and then a date given. Our system ensures intervals are undamaged by constraining date offsets in week-long intervals. Thus, even if the dates are moved by years, there's no loss in day-granular intervals.

The date offset is applied across all records of a single patient uniformly to maintain interval and continuity of encounters. Furthermore, the system is very flexible about handling dates in free text, using as much knowledge as possible to piece together correct, redacted dates. For example, a snippet of a medical note may read: "A surgery was performed in 2005 to correct the issue; on March 4, the patient..." Because the redaction system makes use of the structured fields, it would extract the date of entry for this medical note. Assuming that date is *March 7, 2006*, the system will move forward labeling unspecified years as *2006*, giving a means of differentiating the vague dates *2005* and *March 7*.

4.2.2 Unstructured

The second pass of de-identification also operates over free text, but it does not make use of known information such as the dictionary of names or the dates of an entry. Instead, general attributes of potential PHI are used to locate and remove sensitive data. Email addresses, phone numbers, mailing addresses, and medical case numbers are located through common regular expressions. ZIP codes are retained because they are not considered PHI and can be useful for location-based operations.

Unknown entities appear frequently in the text due to other names of people or places being written that are not listed in the dictionary of names. To account for these entities, we attempted two versions of the unstructured protocol. In the first, the well-known text analysis system from Stanford, *CoreNLP*, was used to detect any remaining entities in the text which do not belong to a linked pseudonym [84]. In the second, the Perl-based tool *deid* was used to find remaining entities. *Deid* is predominantly dictionary-based, which gives it a performance bonus versus *CoreNLP*'s more extensive modelling [43]. Regardless of the tool used, all entities are redacted according to their determined type, e.g. $NAME1$ for a person or $LOCATION1$ for a place. Even in the unstructured phase, sequential naming schemes ensure unknown people and places do not become confounded with any other entities.

4.2.3 Complete Pipeline

By the time the pipeline has finished, the text has been run through two rounds of de-identification. First, any useful knowledge is pulled from the data in the EMR to build a dictionary for rule-based redaction that preserves roles. Second, operating without any knowledge, a set of regular expressions and more sophisticated entity recognition methods are employed to clear other sensitive data without adding ambiguity or destroying valuable non-PHI information. The inclusion of a general tool like *CoreNLP* or *deid* in the final part supports more advanced entity recognition than the former set of regular expressions. This allows the complete pipeline to capture almost any potential PHI while still recognizing known entities, particularly those relevant to IOO, or types of entities, such as contact numbers of locations.

4.3 Validation

To ensure the removal of PHI from the data sets, a team of clinicians reviewed a large set of documents from both VCD and HCD. The clinicians were asked to mark any missed PHI and any non-PHI mistakenly labeled as such; by counting all the redacted and non-redacted words, we were able to create confusion matrices for the PHI recognition performance of the complete redaction pipeline. In the following tables, the error rates are defined as:

- *False Negatives* (FNs) - Missed PHI left unscrubbed

- *False Positives* (FPs) - Non-PHI mistakenly labeled as PHI and anonymized
- *True Positives* (TPs) - Correctly identified PHI that has been redacted
- *True Negatives* (TNs) - Non-PHI correctly ignored

As alluded to before, we performed two distinct iterations of redaction initially – one using *CoreNLP* and the other using *deid* with slightly tweaked patterns to improve compatibility. For VCD, we generated data from both iterations and performed validation. However, because *deid* showed a boost in accuracy over *CoreNLP*, HCD only has data for the second iteration using *deid*. The performance benefit is not entirely unexpected, even given *CoreNLP*'s more sophisticated modeling, because *deid* is designed specifically to support redaction tasks. However, the validation results on VCD did give quantifiable support that *CoreNLP* could be removed from the completed pipeline.

4.3.1 VCD

In each of the VCD validation iterations, clinical experts reviewed sets of mixed-length texts. The document classes were:

- **reason** - Free text containing the presenting complaint of the encounter
- **description** - Description of MR entry
- **fulltext** - Free text from the discharge summaries
- **followup** - The call-if or followup subsection at the end of the fulltext

4.3.1.1 First Iteration

In the first iteration of VCD redaction, clinical experts reviewed approximately 250 documents in all. Table 4.2 shows the error rates for each document type according to the definitions given above. Table 4.3 gives several performance metrics.

Table 4.2: Error rates by document type in the first iteration of VCD redaction.

	reason	**description**	**fulltext**	**followup**
False Negatives	5	7	218	13
False Positives	1	1	209	22
True Positives	18	325	6265	640
True Negatives	504	617	127593	16057

Table 4.3: Performance metrics by document type in the first iteration of VCD redaction.

	reason	**description**	**fulltext**	**followup**
Sensitivity	78.3%	97.9%	96.6%	98.0%
Specificity	99.8%	99.8%	99.8%	99.9%
Precision	94.7%	99.7%	96.8%	96.7%

When found, the vast majority of PHIs were case numbers, which were fully not considered in this version of the protocol. In fact, a full 110 of the PHIs reported in the fulltext category are case numbers. On rare occasion, an animal name could slip through if abbreviated or misspelled, while other times parts of addresses were not found and fully redacted. The short texts did not tend to contain PHIs except dates, which were nearly always replaced. The larger free text documents contained more errors, although the followup sections performed quite well. These sections tended to not contain as much PHI, but they were typically replaced when present.

Of the 13 PHIs found in the followup text, 4 were parts of clinician names not being removed, and the remaining 9 were missed animal names. No case numbers, addresses, phone numbers, owner names, or other directly traceable PHIs were found in the followup section.

Table 4.4: Error rates by document type in the second iteration of VCD redaction.

	reason	**description**	**fulltext**	**followup**
False Negatives	0	0	76	0
False Positives	0	0	5	0
True Positives	2	6	3391	91
True Negatives	268	244	75694	3031

Table 4.5: Performance metrics by document type in the second iteration of VCD redaction.

	reason	**description**	**fulltext**	**followup**
Sensitivity	100.00%	100.00%	97.81%	100.00%
Specificity	100.00%	100.00%	99.99%	100.00%
Precision	100.00%	100.00%	99.85%	100.00%

4.3.1.2 Second Iteration

Experts reviewed about 120 documents for the second iteration. Table 4.4 and Table 4.5 show the error rates and performance metrics, respectively.

In this version of the protocol, no PHIs were found in the reason, description, or followup fields in the sample of 122 documents. The full document text continues to carry some challenges. Only 1 instance of a patient name was found out of all 76 FNs and a total of 79,085 correct redaction determinations. Overall, this version of the pipeline showed a marked improvement in sensitivity, specificity, and precision over the previous version.

4.3.2 HCD

Based on the increase in performance between the first and second runs of VCD redaction, we elected to use only the latter pipeline for HCD. There are not as many text documents in the hospice EMR, so only clinical notes were used. Subject matter

Table 4.6: Error rates for the HCD redaction considering both all names and the pure HIPAA-defined PHI.

	All names, including clinicians	**HIPAA-defined PHI**
False Negatives	170	29
False Positives	250	250
True Positives	1478	1478
True Negatives	51451	51592

Table 4.7: Performance rates for the HCD redaction for all names, including clinicians, and true PHI.

	All names, including clinicians	**HIPAA-defined PHI**
Sensitivity	89.68%	98.08%
Specificity	99.52%	99.52%
Precision	85.53%	85.53%

experts read through 500 notes to generate the error rates shown in Table 4.6 and the performance metrics in Table 4.7.

From a very rigorous measure, which includes any names as possible PHI, the system had 170 missed out of 51,621 words. This is likely due to the difference between VCD and HCD text. In VCD, most of the text is client-side communication, and the names and other PHI used originate elsewhere in the patient record. In HCD, all of the text belongs to clinical notes. These are different in structure than client communication, written more informally with abbreviations and references to names, entities, and PHI not anywhere else in the record.

However, the definition of any name being PHI is beyond the requirements of HIPAA since clinician names are not generally considered PHI. These are instances of clinicians not being listed elsewhere in the file. The measure based strictly on the HIPAA definition of PHI cuts down dramatically on false negatives. We were able to identify for certain over 100 instances of clinician names because of titles like LVM, RN, MD, etc. Of the remaining 29 false negatives, the majority referred to ages over 89 years old and some were ambiguous names that may either be clinician or patient.

5. ANALYSIS OF THE STRUCTURED RECORD

After completing data pre-processing, we began to investigate the structured part of the patient files for outcomes-oriented information. The majority of the analyses discussed in this chapter relate to the VCD set of data, but they are not tied to veterinary care due to the generality of the features and algorithms. As with the pre-processing pipeline, all of the remaining work is done in Python through built-in functionality and third-party libraries.

Revisiting the competency questions from Section 3.1, we know that the first question is answered directly by grabbing fields like patient name and attending clinician from the EMR; Appendix A contains the full list of these fields. The remaining questions need further exploration. To answer the second question, we must be able to relate encounters, which we refer to as "episodes" (Section 3.3). Encounters themselves are saved as appointments, so we set about generating features of encounters that could describe their "relatedness," establishing episodes.

5.1 Features of Encounters

Without delving into the free text, we wish to use simple fields in the data set to generate features. One of the most obvious features is the reason for the encounter. This is typically given in the form of a presenting complaint at the start of the encounter, although there may also be a medical diagnosis associated with the visit. In VCD, we have both.

5.1.1 Diagnosis Codes

Technically, VCD has diagnosis codes and problem codes. The former are codes defined in the Veterinary Medical Databases[1] (VMDB), and the latter are defined internally by VMIS tables. VMDB codes are themselves an extension of the standard SNOMED coding (formerly, short for the Systemized Nomenclature of Medicine but currently a brand name) developed by International Health Terminology Standards Development Organisation (IHTSDO) [85]. While clinical experts on our team did create a mapping between the VMDB codes and problem labels, it has not yet been used for encounter features. We'd first like to build a better understanding of the diagnosis codes themselves. Ideally, this effort could be augmented by the mappings to problems in the future, or perhaps more importantly, the problems could be augmented by this effort. For now, we focus on how to build a feature vector from the diagnosis codes and what the current feature space looks like.

5.1.1.1 Vector Representation

The codes are themselves already similar to a vector. They are 9-digit alphanumeric identifiers. Some structure is inherent in the design, and thus, even a simple vectorization method will incorporate meaning into the representation. To provide a concrete example, Table 5.1 shows several diagnosis codes. The leading digit is most significant and indicates that all codes beginning with a '9' are tied to neurology. The first two are closely related, as seen by the similarity in their codes, while the third may fall under the same broad category but associate with a different part of the body. The fourth shows that codes starting with an 'X' fall under an entirely different category, in this case ophthalmology.

[1] https://vmdb.org

Table 5.1: Sample VMDB diagnosis codes showing similarity among different code groups.

Code	**Explanation**
98118451D	NEUROFIBROMA NERV ROOT
98118Y00D	NERVE ROOT TUMOR D
982460000	SYNCHRONOUS CONTRACT HRT/DIAPH
X13001400	CONGN HYPERPLAS SCLERA

After some discussion, it was decided that we should make use of this structure. The first-pass attempt is based on converting the base 36 (10 numeric + 26 alphabet) codes into base 10. Once we have the base 10 vector, it is easy to plot and visualize codes and compute weighted distances between them to preserve digit significance.

Python facilities this conversion with its built-in *int()* function; *int('98118451D',36)* will tell the interpreter to take the string representation of the code and interpret it as a base 36 number into a base 10 integer. The result is 26019153358465. This is unsatisfactory, however, because the numbers are very large and do not take into account the structure of VMDB. For instance, consider the following example codes:

$int('400000000', 36) = 11284439629824$

$int('4ZZZZZZZZ', 36) = 14105549537279$

$int('500000000', 36) = 14105549537280$

In truth, because of the way VMDB groups categories, 400000000 and 4ZZZZZZZZ should be more closely related to each other than either would be to 500000000, but because we just converted the numbers linearly without care for the significance of the digits, 4ZZZZZZZZ and 500000000 are represented as being much closer than they otherwise would be (a difference of 1). The solution is to apply weighted distances using the digit significance. To do that, we have to apply conversion digit-by-digit, as we see in Table 5.2 for the code 4ZZZZZZZZ.

Table 5.2: Sample VMDB code conversion at the digit-level.

Base 36	4	Z	Z	Z	Z	Z	Z	Z	Z
Base 10	4	35	35	35	35	35	35	35	35

With this process, when we compute the distance between codes with a weight for each digit, we can more accurately represent the groupings in VMDB. In fact, if we give each digit a base 10 order of magnitude increase over the previous place, so that the leading digit would have the weight 10^8, the difference between 4ZZZZZZZZ and 500000000 would be $1 * 10^8 + 35 * 10^7 + 35 * 10^6 + ...$ The difference between 4ZZZZZZZZ and 400000000 would be $0 * 10^8 + 35 * 10^7 + 35 * 10^6 + ...$ Thus, 4ZZZZZZZZ will be 10^8 closer to 400000000 than it would to 500000000. This is just an example of how using a weighted distance better captures the intent behind VMDB.

5.1.1.2 Clustering Diagnosis Codes

Given a method for vectorizing codes, we wanted to see how much structure we could capture simply by using VMDB directly. This is just an investigation because we'd like to incorporate more meaning into the diagnosis code vector. For instance, down the road we plan to embed co-mordibity of diagnosis codes so that even codes appearing outside the same category could be grouped closer than they otherwise might be if they occur together frequently, e.g. a food allergy with skin irritation.

To determine how related the diagnosis codes are to one another from this vectorization method, we converted all codes into vectors, performed Principal Components Analysis (PCA) down to 2 dimensions [86, 87], and then plotted the findings on a scatter plot. The results are shown in Figure 5.1. The color bar shows the colors for the text category labels provided by the clinicians.

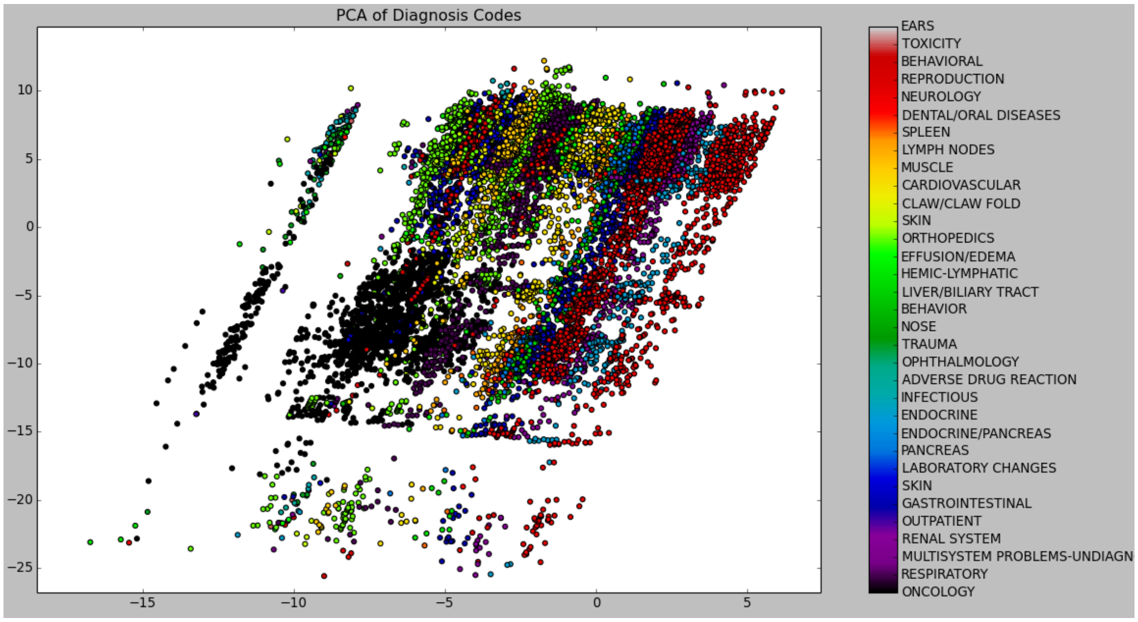

Figure 5.1: Principal Components Analysis of diagnosis codes.

The differences among the classes is very noticeable. Though they're not separated by as large a gap as we might hope, they are clearly in distinct groups. The striations along the diagonal direction demonstrate that many of the codes with leading digits are of the same class, and the digits progress incrementally. Therefore, as we had hoped, there is an existing structure to the VMDB coding which can be easily captured in a simplified base 10 form.

5.1.2 Presenting Complaint

Diagnosis codes describe the prescribing provider's reason for an encounter, but there is also the reason given by the patient or observing caregiver. In VCD, these are presenting complaints, single lines of text explaining the problem from a layman perspective. These are not as inherently structural as VMDB codes and are only use for surface similarity comparisons. We elected to use the common cosine similarity method from the field of Information Retrieval (IR) [88]. Cosine similarity, very similar to the Jacaard Coefficient [89], measures the number of elements in both sets divided by the total number of elements in each set. It is essentially a ratio of the overlap between two sets; in IR, this is a quick way to determine the relatedness of two documents. The formal representation is shown in Equation 5.1.

$$similarity = cos\theta = \frac{A \cdot B}{\|A\|\|B\|} \tag{5.1}$$

5.1.3 Medications

Although the list of prescribed medications may be more close to interventions, we use medication lists as features of encounters as well. The reasoning is that common treatment are given for similar types of problems. Rather than a detailed exploration, like presenting complaints, the string list of medications is compared using cosine similarity between two encounters.

5.1.4 Age

Consultation with subject matter experts led to the recommendation to include a time-based feature in identifying episodes of care. While the pre-processing pipeline does preserve time intervals, even though dates themselves are shifted, a much better solution is to use patient age. Age is already a scalar value with a standard minimum, maximum, and average across patient cohorts, like breeds in veterinary data.

Another benefit to age is supporting direct comparison between patients. Using only time intervals, one wouldn't be able to find the average age of cancer onset, for instance, but age encodes this type of query into the encounter features.

5.2 Episode Identification

Using encounter features, we can separate multiple encounters into groups based on their similarity. Because two features, presenting complaint and medications, are

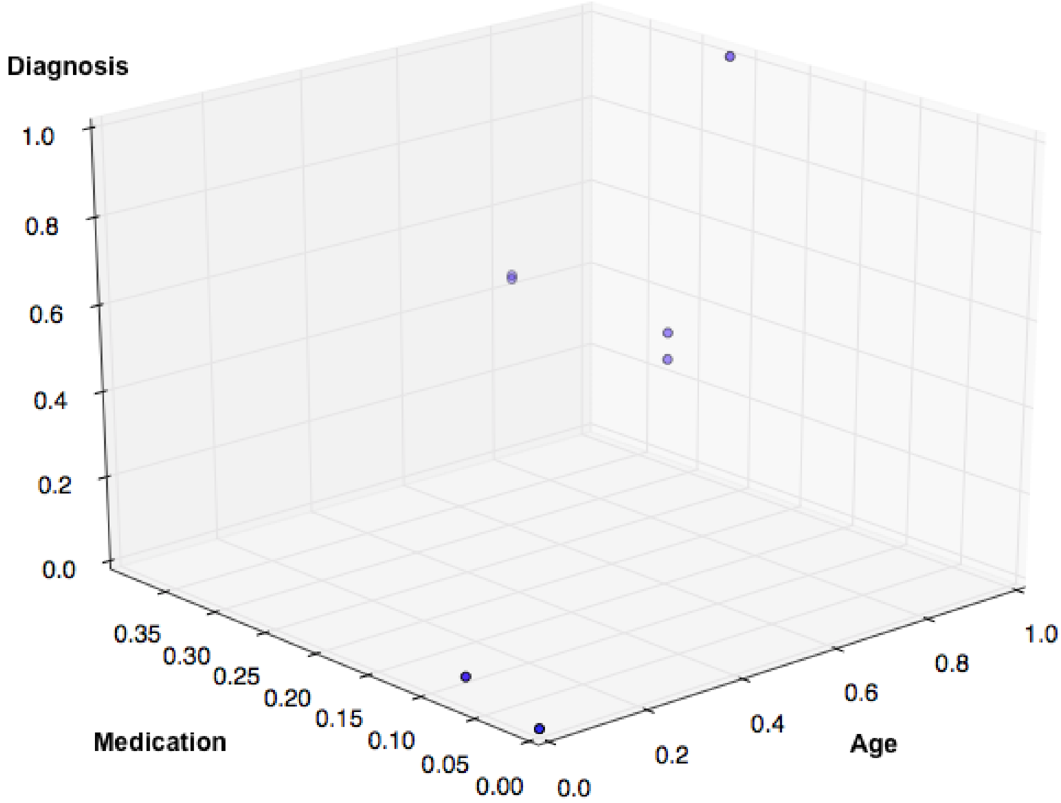

Figure 5.2: A scatter plot showing multiple encounters relative to each other in a vector space comprised of age, medications, and diagnosis codes.

based on differences between sets, we are able to generate a pairwise-similarity, or distance, matrix between a group of encounters. Several machine learning techniques would be appropriate for identifying episodes of care. We use hierarchical clustering because it gives a consistent grouping of encounters across all patients for a single patient's distance matrix [90].

Figure 5.2 gives a case study in the form of a single patient's encounters plotted relative one another. This plot does not include the presenting complaint similarity axis, and all other axes are normalized between 0 and 1. For this patient, there are three episodes of care identified by human experts, and when plotted according to their similarity, we see that most clustering techniques would select three groups.

In preparation for the evaluation of hierarchical clustering and episode identification, clinical experts on our team provided an episode key. This list consisted of groupings of encounters for 285 patients. The criteria was based largely on human expert intuition. Discussion revealed that time, presenting complaint, and diagnosis were heavily used.

5.2.1 Baseline Using Age

We see in Figure 5.2 that the age axis plays a significant role in separating clusters. Human experts suggested that age was the single greatest criteria for grouping encounters during discussion. For a trial attempt at grouping encounters and to form a baseline, we first use only age to generate the distance matrix for hierarchical clustering. We use the third-party Python library SciPy[2] to perform hierarchical clustering using the *fcluster* function. It includes a parameter called *max_d* that specifies the distance to allow before cutting.

Rand index is used for evaluation, being one of the more reliable measures of clustering because it considers true positives and false negatives [91]. In our case, SciPy's *adjusted_rand_score* function can generate the results. To make a compatible list of labels, all the clusters created by the algorithm and experts were sorted according to their encounter numbers. Then, cluster numbers were assigned based on that ordering. Once labels have been assigned to every encounter, they can be compared directly and the final results obtained by averaging over all 285 samples, although this does introduce high variability to the results.

[2] https://www.scipy.org

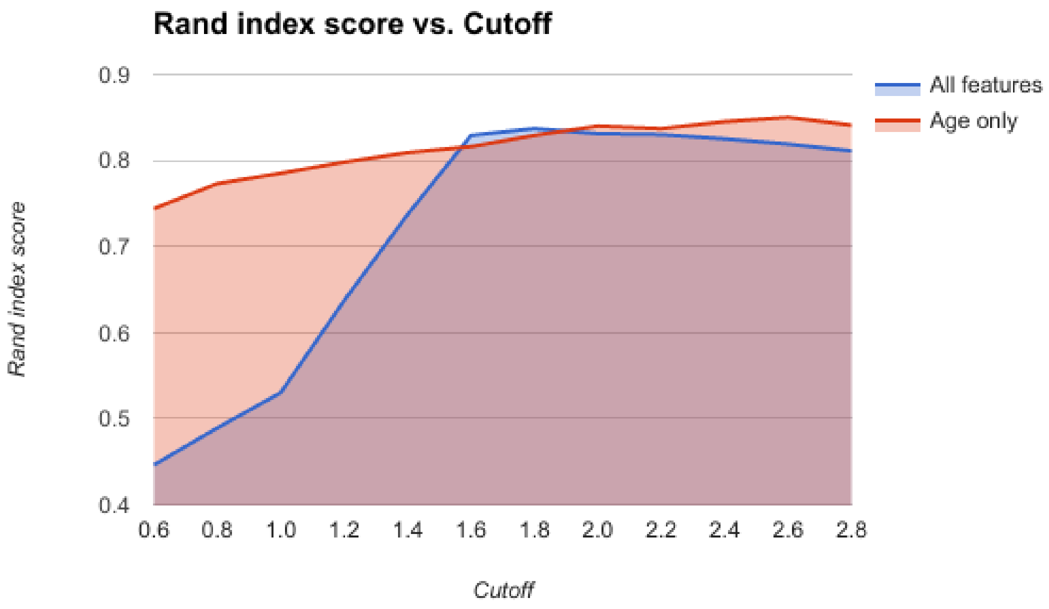

Figure 5.3: Area chart comparing the baseline model against the multidimensional one for hierarchical clustering of encounters into episodes.

5.2.2 Combining Features

After developing the baseline, we used SciPy's ability to create multidimensional linkage graphs for hierarchical clustering. A cutoff still needed to be selected, so we ran the algorithm at multiple cutoff points and selected the best performance.

Figure 5.3 shows the performance of the combined model at multiple cutoff points against the baseline. Surprisingly, the baseline method of using only age performs exceedingly well against a more comprehensive model. In particular, it is less susceptible to forming poor clusters at low cutoffs, which tend to divide every encounter into its own episode and ignore similarity if too small. Both improve with a higher cutoff, until eventually yielding diminishing returns before the scores begin to drop, albeit more slowly. We can conclude that a higher cutoff is better than a lower one because fewer episodes is more likely than many episodes for most patients. At around 1.6 cutoff, the model incorporating all features is slightly better with a rand index score of $83.9 \pm 26.1\%$ versus $81.6 \pm 29.1\%$ using only age. Peak performance is obtained at 2.6 for the baseline with $85 \pm 26.4\%$ against $81.9 \pm 29.2\%$ using all features.

Even though none of these scores is perfect, with a rand index of 84%, we can reasonably segment episodes of care algorithmically, a partial answer to the second competency question of IOO. The most significant takeaway of these findings is that a simple model based solely on age truly may be enough to identify episodes of care. Other algorithms may perform even better, such as logistic regression comparing only the current encounter with the previous as opposed to clustering on all encounters at once. Leaving that further investigation for later, we now turn to the remaining key ontological elements, their competency questions, and the associated research questions: interventions and outcomes.

5.3 Interventions and Outcomes

IOO's third and sixth competency questions are concerned with locating interventions in the medical record and determining if a change has occurred. Our initial approach to interventions is to consider only prescribed medications. With this definition, not only are the interventions recorded, but changes are a simple matter to detect by matching medicine names across encounters. In the current system, dosage and formulation are ignored.

We performed qualitative analysis on this approach with clinical experts and found that initiating, continuing, and discontinuing a therapy are trivial to detect and classify. Alterations are more complex because humans judge the content to determine the type of alteration. That is, many drugs may be discontinued, but if a new one is added with more dangerous side effects, the human expert considers the addition to be the most relevant alteration to the care plan. Such information is not present in the structured record and must be pulled from elsewhere since the computer doesn't have any intuitive way of evaluating the relative importance of the medications.

Attempts to identify outcomes in the structured record have proven largely unsuccessful. In VCD, outcomes are not captured at all except in cases like death. In HCD, outcomes are stored more coherently, with an associated intervention and goal code. Unfortunately, almost all outcomes-oriented details are found in the free text documents. With the limitations of structured analysis for interventions and outcomes in mind, we turn to the very large field of NLP to mine information in the medical texts.

6. ANALYSIS OF THE UNSTRUCTURED RECORD

In VCD, there are several types of documents: emails, clinical notes, diagnostic results, and more. The bulk of free text comes in the form of patient discharge summaries, comprehensive write-ups given to the caregiver at the time of release. These are sectioned documents with patient history, descriptions of the presenting problem, treatments given, and the recovery care plan. In HCD, the only free text type of document is a clinical note.

In both cases, identifying potential outcomes has proven difficult from the structured record. However, in documents like discharge summaries, prescribing providers do explain outcomes to some extent. Certainly, undesirable side effects of the treatment are stated as a caution. In other cases, the time until recovery is likely to be given. This type of knowledge is lacking in the rest of the patient file, but it is the form of information needed to answer the remaining competency questions posed by IOO.

6.1 Segmenting Discharge Summaries

Discharge summaries follow a template. Because they are released to patients, they must be well organized and easy to understand. We leveraged this attribute by identifying the standard headers and formats in VCD discharge summaries and auto-segmenting them. The most common sections we could automatically segment are listed below.

- History - A brief background on the patient leading up to the current encounter
- Physical examination - The results of the exam given at admission
- Diagnosis - The likely problem that led to the encounter

- Treatment - Actions taken to resolve the problem
- Exercise - Recommended activities or restrictions to perform at home
- Diet - Any recommended alterations to food and drink intake after discharge
- Followup or Call if - Information about side effects or complications which require a subsequent encounter

These are all written in natural language, but they are rich with potential for IOO tasks. In particular, based on our focus groups with subject matter experts, the latter sections were chosen as most likely to contain interventions and outcomes; these are "treatment," "exercise," "diet," and "call if." HCD clinical notes are not as easily segmented, so in order to address the seventh competency question from Section 3.1, we sought ways to programmatically segment the text based on content rather than format.

Using Python's scikit-learn module[1], we built a segment classifier to determine the type of therapy based on the words in the section [92]. Bag-of-words modeling, looking only at the collection of words in a document to decide its type, is commonly used in NLP and IR [93]. Scikit-learn contains a bag-of-words vectorizer with TF*IDF scoring. TF*IDF, Term-Frequency*Inverse-Document-Frequency, calculates a weighted score of a word based on its frequency [94, 95, 96]. In Equation 6.1, TF*IDF is the product of term frequency, the count of a given term in a document over all terms in the document, with inverse document frequency, the total number of documents over the number of documents with the term.

[1] http://scikit-learn.org

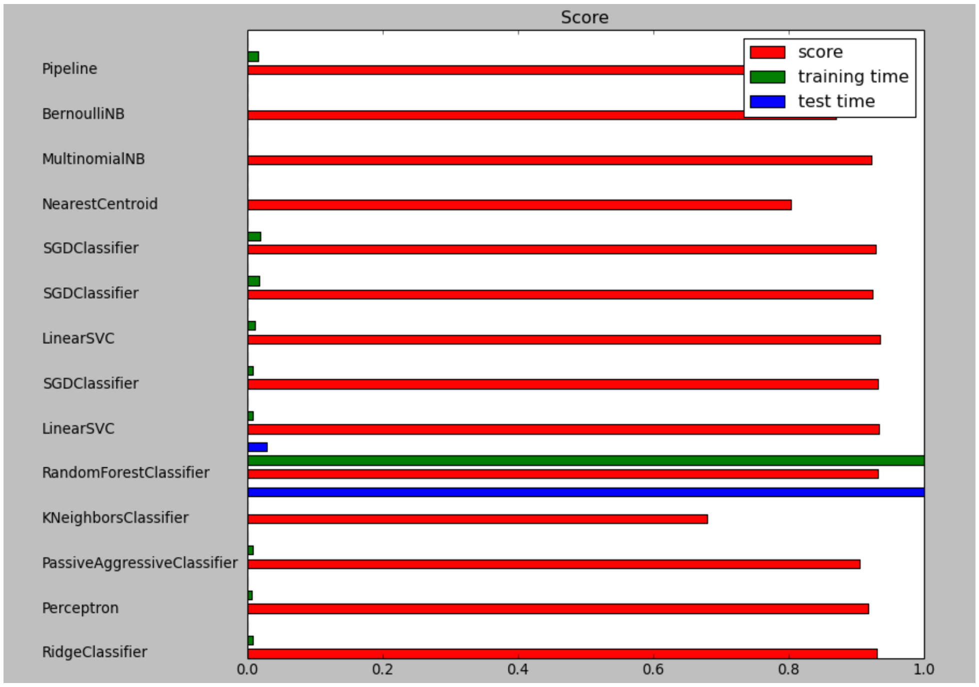

Figure 6.1: Different classifier performance on labeling three classes based on bag-of-words.

$$TF * IDF = \frac{f_t, d}{\sum_{t' \in d} f_{t',d}} * \log \frac{N}{n_t} \tag{6.1}$$

Initially, we test classifying over three different sections, a subset of the VCD documents – 34,577 of treatment, 32,601 of exercise, and 33,068 of diet containers. With TF*IDF scoring and bag-of-words, we tested a large collection of classifiers. Figure 6.1 shows labeling accuracy for multiple classifiers. With only three classes, the accuracy is quite high overall, and the words in the containers are fairly distinct. The top ten words in each container are given in Table 6.1

By way of visualization, Singular Value Decomposition (SVD) was used to create a three dimensional graph of the words in each segment. SVD, similar to PCA, reduces the high dimensional vector space of the TF*IDF weights to only three dimensions

Table 6.1: Several segments of discharge summaries and their 10 most frequent words.

Segment	**Top 10 Words**
Treatment	change, diet, time, continue, exercise, food, usual, iv, weight, today
Diet	normal, diet, continue, regular, current, exercise, feed, routine, eat, resume
Exercise	pace, set, let, allow, exercise, continue, activity, restrictions, level, home

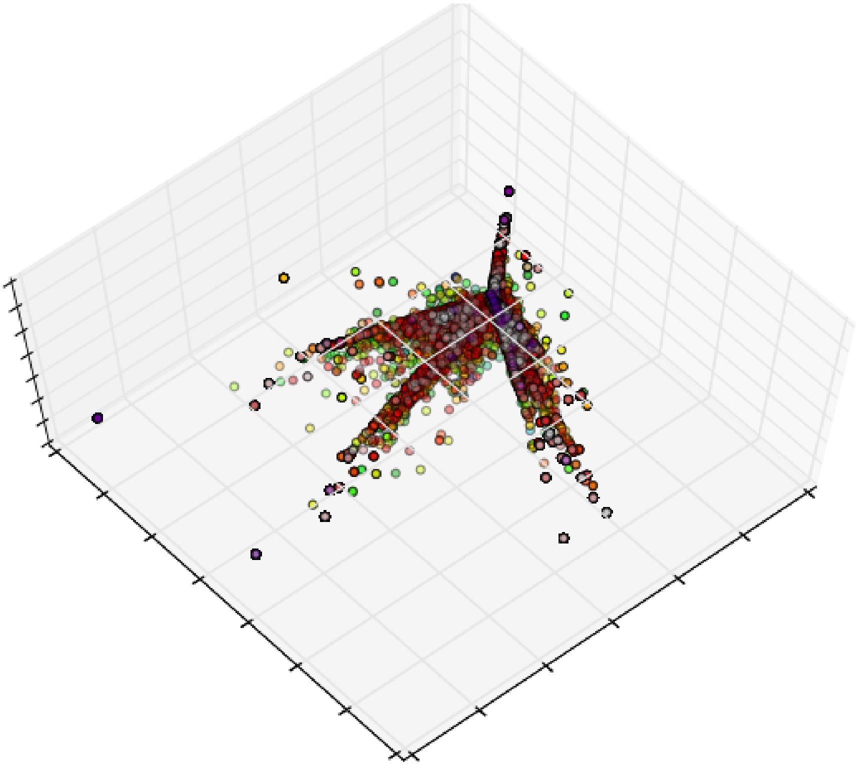

Figure 6.2: Bag-of-words scoring vectors decomposed to a three dimensional representation.

[97]. Figure 6.2 indicates that the classes remain relatively separable even at this lower dimensional representation as there are several distinct groups of words.

Another visual tool for characterizing the text data is a word cloud. Figure 6.3 is a decorative word cloud, displaying the relative weight of each of the most frequent words in the VCD documents. We also generated clouds to help visualize the words in the treatment, diet, and exercise segments. These were originally created to aid discussion with subject matter experts and help identify keywords for types of interventions. Appendix B includes all of these visuals.

6.2 Word Embeddings

To power more sophisticated NLP algorithms, we developed word embeddings in favor of bag-of-words with TF*IDF scoring. Like with our earlier goal of vectorizing VMDB diagnosis codes, a vector representation of the vocabulary of words is needed. This conversion is more challenging than working with codes because words contain so much variety, but many researchers have developed approaches that are widely used.

In order to prepare our data for clustering, all VCD discharge summaries and HCD clinical notes were downloaded and concatenated to form single input files from each domain. We returned to our role-based redaction while normalizing text to serve as training corpora. Specifically, instances like $Clark_{CAREGIVER1}$ as we saw back in Table 4.1 were replaced by *caregiver* in the text. Addresses, case numbers, prescribing providers, and other redacted entities were replaced with their recognized role. From a semantic perspective, this allows the meaning of the terms to be retained, also preserving the context of nearby words, without needing to revisit any original data from the redacted source – an excellent argument in support of role-preserving redaction.

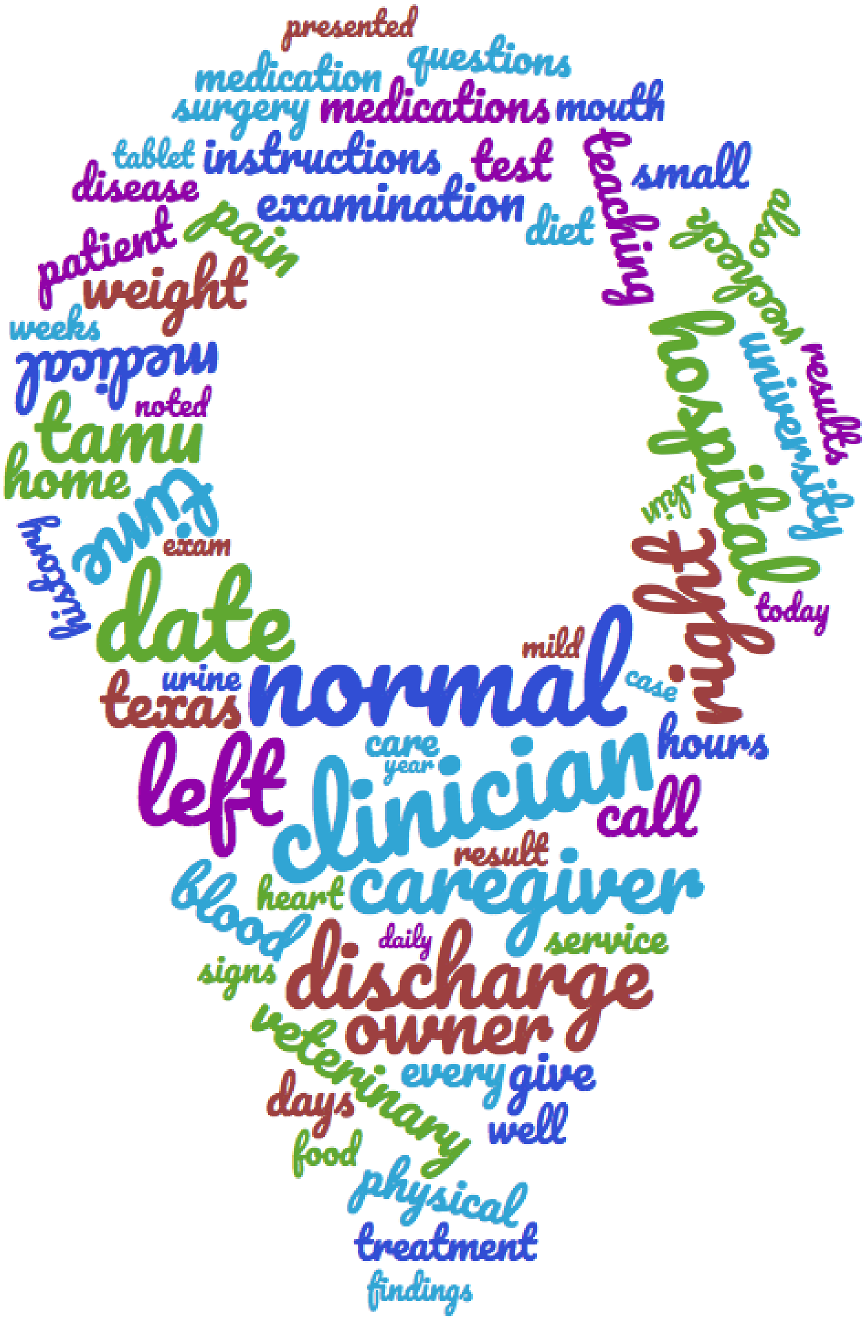

Figure 6.3: Stylized word cloud showing weighted significance of the most frequent words in all the VCD documents.

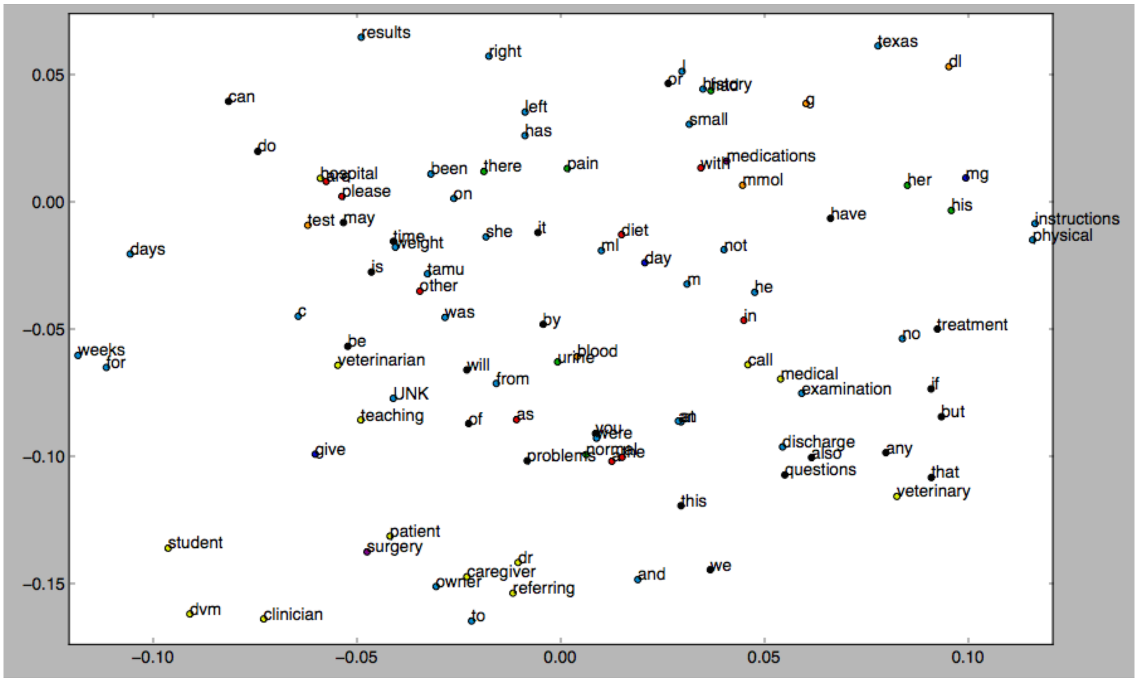

Figure 6.4: Word2Vec embeddings in two dimensions using PCA for the most frequent 100 words in VCD; we see that some semantics are retained through word proximity even at this crude level.

6.2.1 Word2Vec

One of the latest techniques for vectorizing words is the *Word2Vec* algorithm [98, 99]. *Word2Vec* builds large dimensional vectors to embed (hence, word embeddings) the semantics of a word into a number. These vectors are so sizable, typically hundreds of axes, because words have very complex meanings and require a significant vector space to retain their meaning. The location of the word in the vector space imparts some of its meaning, and nearby words are semantically similar. In fact, one of the interesting aspects of Word2Vec is that mathematical operations like addition and subtraction are applicable, as well as analogies. Using the Word2Vec representation, an equation like $king - man$ should return $queen$.

Word2Vec uses a window of words on either side of the current word to build an expectation-maximization model [100]; that is, the system attempts to guess the most likely word given the surroundings and learns the semantics of that word in the process. We used the open source TensorFlow[2] toolkit with its built-in word embeddings generator functions to create our Word2Vec model.

TensorFlow generates 200-dimensional vectors by default to represent each word. Using PCA to reduce these vectors to only two dimensions, Figure 6.4 plots the most frequent 100 words in VCD. Even reduced to two dimensions, some of the semantics are retained through word proximity. For example, "student," "dvm," and "clinician" are all nearby each other; "her" and "him" are next to each other; and so on.

Unfortunately, too much information is lost using PCA or SVD to encode the entire vocabulary. In order to use Word2Vec as a single feature in other models, we applied K-Means clustering to clump large sets of nearby words into semantically-similar groups [101]. To inspect the clusters created by the K-Means algorithm, we placed the entire vocabulary in a K-D Tree, a k-dimensional tree structure which is very efficient for searching across large data sets [102]. Scikit-learn has functions for K-D Trees and K-Nearest-Neighbor (KNN) querying [103], so the nearest words to each K-Means cluster center could be quickly retrieved.

With the ability to locate the cluster centers and nearby words quickly, we tested multiple values for k (the number of clusters) and computed the average sum of squares distance from the centroid to other words in the cluster. As the number of clusters increases, the distance should drop with improving coverage, but, ideally, there would be an "elbow" to the curve when more clusters lead to diminishing

[2] https://www.tensorflow.org

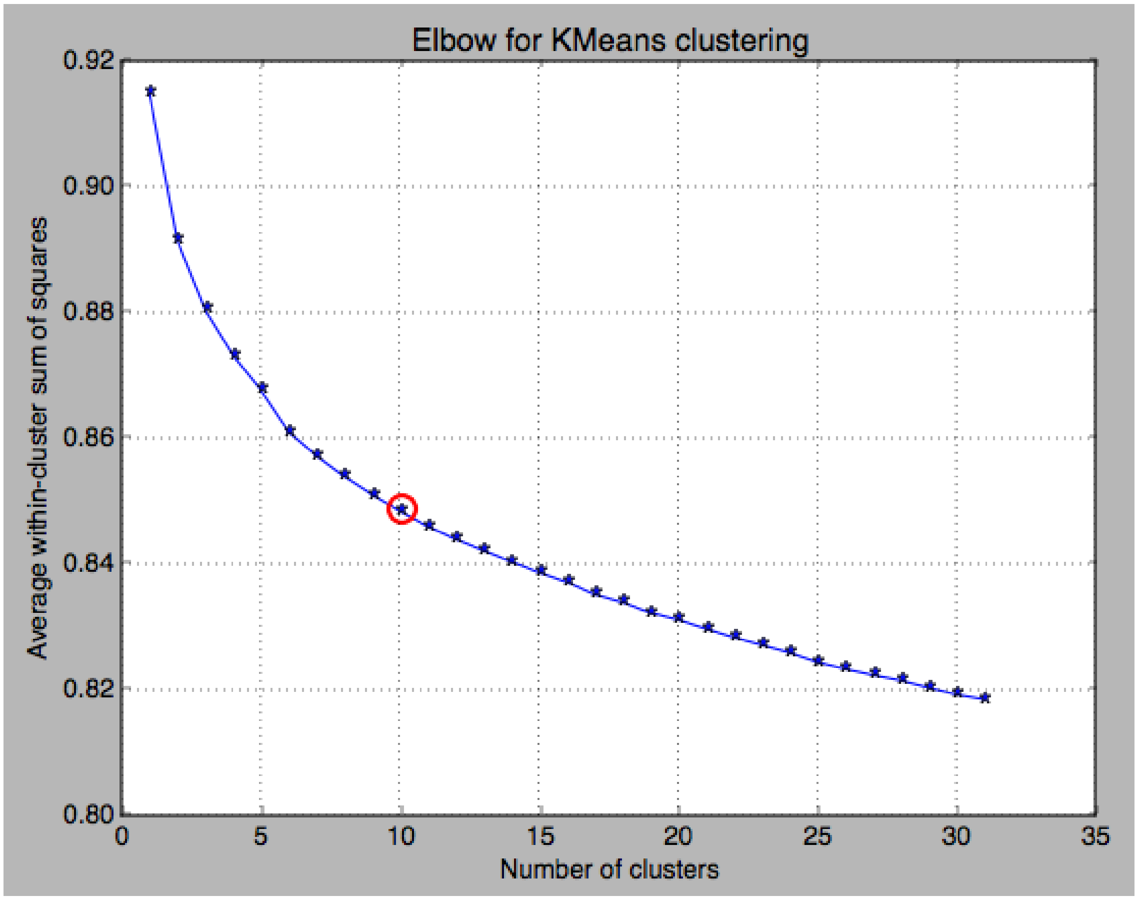

Figure 6.5: Average sum of squares distance within the clusters for different values of k when building K-Means clusters to reference the expansive Word2Vec model.

returns. No prominent "elbow" can be seen in Figure 6.5, but significant gains slow after about a dozen cluster centers. Though slightly subjective in precision, we selected 10 for the value of k for VCD and HCD.

In Table 6.2, the five nearest words to each cluster center using the K-D Tree are given. Although these words have been stemmed and are mostly medical terminology, we can see that the clusters are consistent internally. Clusters 0 and 9 have grouped pseudonyms; cluster 1 contains words relating to conversations about health care; cluster 2 is ways to give medicine; etc. Looking back at Figure 6.4, the word points

Table 6.2: Five nearest words to the 10 cluster centers selected by K-Means in VCD.

#	**Nearest 5 Words**
0	'masonic', 'portofino', 'claremont', 'vicksburg', 'refugio'
1	'talk', 'invested', 'recommendation', 'adjuster', 'valued'
2	'antiparas', 'diar', 'histami', 'antiproto', 'bymouth'
3	'comodones', 'hyperkeratinization', 'edematous', 'evident', 'erythematous'
4	'psammoma', 'intima', 'striation', 'nodularity', 'desmoplasia'
5	'tomaintain', 'scaple', 'washers', 'equidistant', 'slits'
6	'barginear', 'hirschman', 'playford', 'kerfoot', 'flemings'
7	'igloo', 'sneaking', 'digs', 'comforting', 'catnip'
8	'thompson', 'lopez', 'miller', 'johnson', 'jones'
9	'positi', 'hemolysin', 'infeccin', 'secundaria', 'hinchazn'

are also colored according to their cluster label. Not only are “student,” “dvm,” and “clinician” near each other, but they have been assigned the same cluster as well. The same goes for “her” and “him” and others.

6.2.2 Brown Clusters

Another vectorization technique, Brown clustering is a means of grouping words by contextual similarity [104, 105]. The algorithm creates binary cluster labels, adding a new digit as the number of vocabulary words grows. Words are assigned a cluster label based on the words around them and their own n-grams, incorporating some topological similarity which can place words with similar structure, prefixes, or suffixes closer to each other.

Consider the example given in Table 6.3. One can see how the clusters become progressively finer with more binary digits. If we only consider the first two digits, “the” and “chased” would belong in their own group while “dog,” “mouse,” and “cat” would be grouped together. All of the computation is done beforehand; with a dictionary of words and clusters, any granularity grouping can be achieved by choos-

Table 6.3: Sample Brown clusters.

Input Sentences	**Word**	**Cluster**
the cat chased the mouse	the	0000
	chased	1000
the dog chased the cat	dog	1100
	mouse	1110
the mouse chased the dog	cat	1111

ing the cluster size in real time. While not as meaningful a vector space as Word2Vec, Brown clusters are flexible, making them an appealing word representation to have.

We use an open source C++ implementation of Brown clustering provided by Percy Liang[3] [106]. The algorithm ran on the same data as TensorFlow, but because it only uses binary labels instead of high-dimension vectors, the results are stored in a human readable text file. The only consideration when building Brown clusters is the number of clusters to allow. Interestingly, investigative work by Derczynski has shown that approximately 10^3 or 1000 Brown clusters consistently performs well on data sets of varying size, which is the number we chose [107].

6.3 Expert Annotation

To this point, we have not yet discussed the IOO competency questions related to outcomes, four and five, and our efforts to answer intervention-related questions three and six have been partially successful but are ultimately limited by the lack of information in the EMR fields. Following an extensive discussion of text characterization, we now turn to automatic labeling of document terms as a means to mine potential intervention and outcome content from medical texts. Two clinical experts

[3] https://github.com/percyliang/brown-cluster

on our team built a training corpus using the Java-based CLAMP application, which features an annotation user interface [59].

The focus was on finding and labeling interventions and outcomes in the text. Consistent with our slightly-constrained definition that an intervention is a medicine or therapy, they labeled terms according to one of five intervention classes:

- Therapy
    - Change
    - Continue
    - Stop
    - Initiate
- General

They labeled both undesirable and desirable outcomes, along with outcome risks or side effects.

- Unintended Outcome Risk
- Intended Outcome Risk
- Unintended Outcome
- Intended Outcome

In total, they annotated 173 discharge summary segments from VCD and 53 clinical notes from HCD. Inter-rater agreement was evaluated using Kappa scoring, a common method of quantifying the level of agreement between human annotators [108]. It is very similar to other scoring metrics, increasing points when a label is the

Table 6.4: Kappa scores denoting inter-rater agreement for the intervention and outcome corpus.

	Interventions		**Outcomes**	
Iteration	**VCD**	**HCD**	**VCD**	**HCD**
1st	0.6133	0.5378	0.5369	0.6029
2nd	0.7530	0.8004	0.8055	0.8594
3rd	0.9528	0.9727	0.9246	0.9633

same from both annotators and decreasing for disagreement, as in rand index scoring and cluster overlap [109]. Table 6.4 shows the kappa scores for both data sets and label groups across three iterations. By incrementally evaluating the agreement, the final training corpus has very high internal consistency.

6.4 Computer Annotation

We use a Conditional Random Field (CRF) model to perform automatic labeling of interventions and outcomes in the medical text. The specific implementation is python-crfsuite[4], a Python library binding to the C-based CRFsuite fast implementation built by Naoaki Okazaki[5] [110, 111]. All of the text parsing is done with Python's Natural Language Toolkit[6] (NLTK) [112].

6.4.1 Features

The CRF models are trained over the expert-annotated data, learning over an assortment of features. Word embeddings are used to capture semantics. Both the Word2Vec cluster label, which allows the full 200-dimensional vector space to be represented in a single number by grouping only similar words, and the Brown binary cluster labels are used.

[4] https://python-crfsuite.readthedocs.io/en/latest/

[5] http://www.chokkan.org/software/crfsuite/

[6] http://www.nltk.org

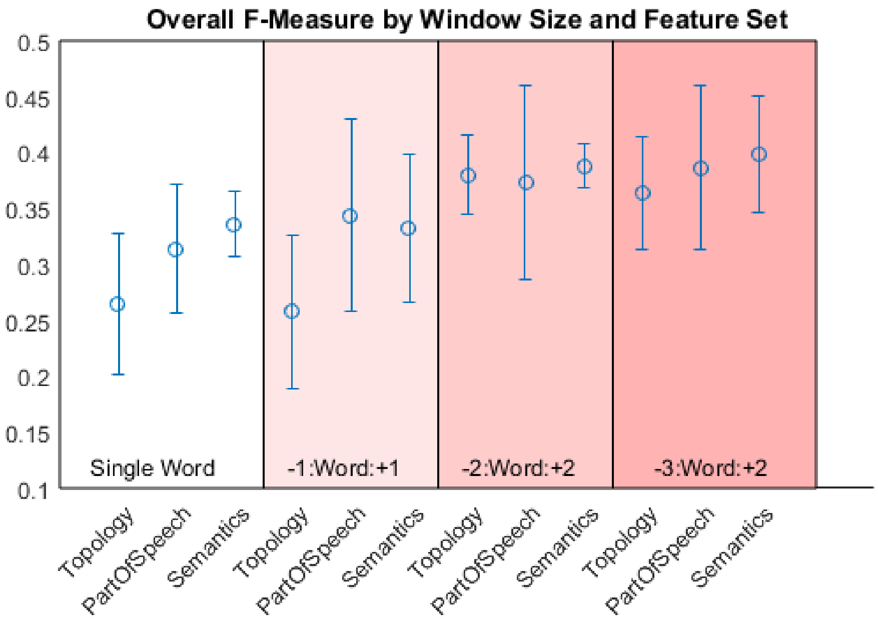

Figure 6.6: The overall F1-score for the CRF on VCD interventions according to different window sizes and feature sets. Window sizes vary from the single word to the past three words and next two. Feature sets consider only topology, topology + part of speech, and topology + part of speech + semantic clustering.

However, we need more than just the semantic grouping to build a good CRF. A word's part of speech (POS) is very important in NLP, supporting tasks like chunking and entity recognition. We use NLTK's built-in POS tagger to generate this feature. Likewise, NLTK has a stemming component which can trim all words to their base form; we apply the popular Porter stemming algorithm [113].

There is also an undeniable importance to the topology of a word. That is, words that have the same structure may mean different things but can be used in similar ways. Consider how many words ending in "ing" represent an action or state. Words with the suffix "ology" often refer to a field of study. Thus, we take the last three characters of a word to represent the suffix. Python supports easily grabbing other

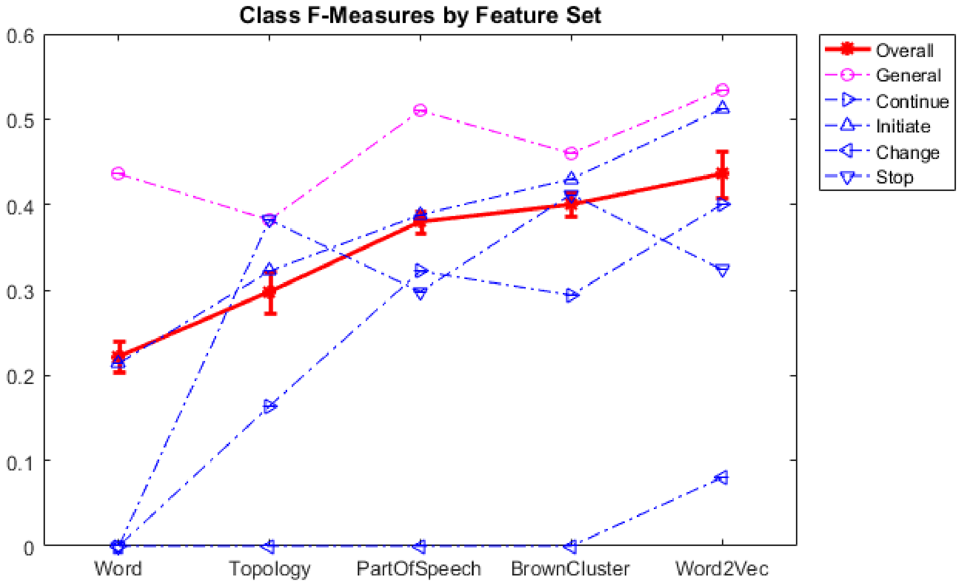

Figure 6.7: Individual class F1-scores in the -3:+2 word window. 'Word' uses only the word stem, while 'topology' includes other aspects like the suffix. 'PartOfSpeech' remains the same as before, but 'Brown' and 'Word2Vec' have been split into two stages from the former 'Semantics' label.

features, such as if the first letter is uppercase, the entire word is capitalized or in title case, or if it is a number.

With a large pool of available features, we conducted several experiments to evaluate the performance of the CRF using different combinations of features and window sizes. Figure 6.6 shows how the overall F1-score changes with window size and feature sets, with mean and standard deviations computed from five iterations at each point. The feature sets consider only topology (e.g. the word stem, suffix, or character case), topology + part of speech, and topology + part of speech + semantics (e.g. Brown and Word2Vec clusters. While parts of speech and semantics improve performance, the most noticeable performance boost comes from growing the window from -1:+1 to -2:+2.

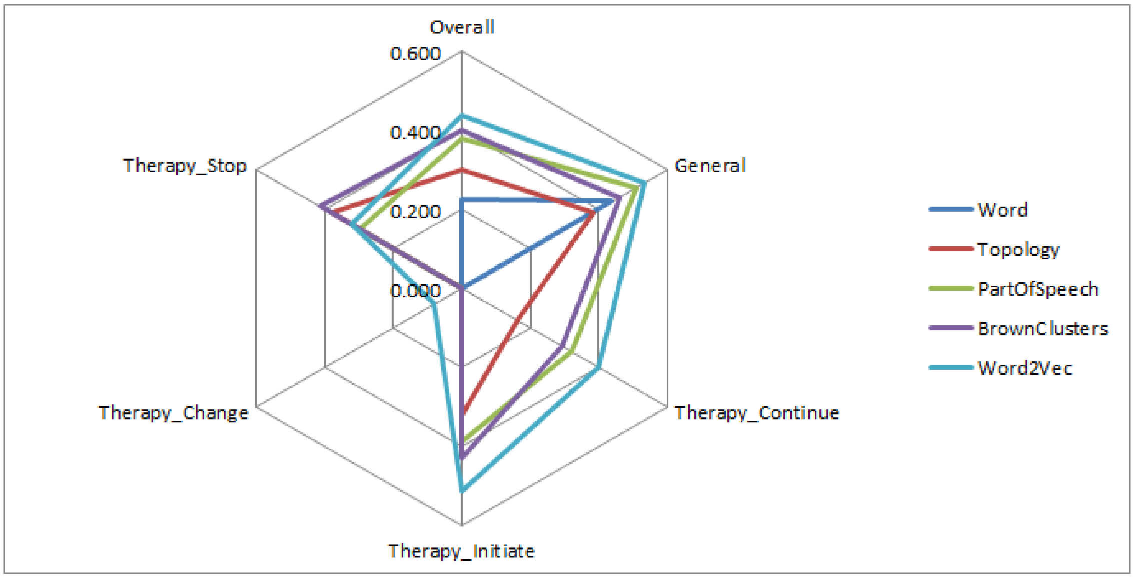

Figure 6.8: Radar chart showing F1-scores by class according to the feature set.

We selected the -3:+2 window size with the full feature set based on the overall F1-score. Figure 6.7 provides a more detailed look at how the feature set affects individuals classes. The labels remain the same as before with only 'Word' being broken out of 'Topology' as a special case of only the word stem, and 'Brown' and 'Word2Vec' being separated from the single 'Semantics' label. The features are still cumulative, so all preceding ones are included in the 'Word2Vec' iterations. As we see, there is quite a bit of variability across the classes, but the overall F1-score does maintain a steady climb as the features become more rich.

Figure 6.8 shows another visualization of this information. Because 'Word2Vec' has the largest area, it gives the most coverage across all classes and represents the final feature set. Unfortunately, detecting therapy changes is weak in all cases. This seems largely due to the difficulty of differentiating between a changed or continued therapy, which can be a subtle distinction to a human expert as well. We will see this particular issue again in the example given at the end of Section 6.4.2.

Table 6.5: Features of each word in the CRF word window; the number is the distance from the current word (0). Key: Brown - Brown cluster, POS - POS tag, IsCap - Starts with a capital letter, IsUpper - Word is uppercase, Stem - Porter stem of the word, Suffix - Last 3 characters, W2V - Cluster number in the Word2Vec vector space

	-3	**-2**	**-1**	**0**	**+1**	**+2**
Features	Brown POS Stem	Brown POS Stem	Brown POS IsCap Stem W2V	Brown POS IsCap IsUpper Stem Suffix W2V	Brown POS IsCap Stem W2V	Brown POS Stem

Regarding the apparently low F1-scores, recall that the CRF is labeling terms in text amid thousands of non-relevant words. Even though these scores are sub-optimal, the system performs very well at distinguishing relevant and non-relevant text, an important difference from an approach like majority classification. Table 6.6 will demonstrate this point for this CRF.

Table 6.5 shows the complete, final set of features in the word window from the current word. Word 0 is the central word, word -1 the previous, word +1 the next, and so on. Notice that the richness of the features varies with the distance from the current word. This is to avoid selecting too many features, given that there is only data from two annotators, while also preserving the contextual features of each word. As we saw in Figures 6.6 and 6.7, improvements diminish as the word window and number of features increase too high because the limited size of our corpus; availability of data must always be a consideration when determining the final feature set.

Table 6.6: Performance metrics for VCD CRF trained to extract intervention/therapy labels.

Label	**Precision**	**Recall**	**F1-score**	**Support**
General	0.85	0.67	0.75	33
Therapy::Change_Therapy	0.00	0.00	0.00	1
Therapy::Continue_Therapy	0.75	0.47	0.58	19
Therapy::Initiate_Therapy	0.75	0.63	0.68	43
Therapy::Stop_Therapy	1.00	0.50	0.67	8
Average / Total	0.79	0.60	0.68	104
Unlabeled Text	0.98	1.00	0.99	1303

6.4.2 Evaluation

By splitting the annotated data into 85% training and 15% testing, we were able to compute several performance metrics for the CRFs, broken down by classes of labels. For VCD, there are two distinct CRFs, one for interventions and the other for outcomes. HCD uses a single CRF with combined labeling.

Table 6.6 shows generally consistent performance when labeling therapies, with an average F1-score of 68%. This score and the others we present may appear low, but as briefly discussed before, because the CRF must label words in large text documents, there is huge potential for error, indicated by the high precision scores but lower recall. To emphasize this consideration, all unlabeled text can be treated as its own class to generate error metrics. This is reported in the last row under "Unlabeled Text," which shows the CRF is highly competent at detecting non-relevant text with an F1-score of 99%. A majority classifier running on this data would mark all text as non-relevant and could achieve high scores as well, but it would be unable to locate any relevant text. Our CRF maintains a strong ability to differentiate labeled and unlabeled text.

Table 6.7: Confusion matrix for the VCD Intervention CRF based on human validation.

	Labeled +	**Labeled -**
Actual +	363	53
Actual -	40	36796

Certain classes perform lower than others; this is in part due to availability of data. Notice that there is less support for the *Change_Therapy* and *Stop_Therapy* labels. These do not seem as common in discharge summary text because changes to the medication or discontinued therapies no longer need to be included with instructions for discharge and followup.

We also asked our clinical experts to perform human validation of this CRF before creating the other models. The CRF was used to generate a set of 100 new intervention-tagged files, and the clinicians were asked to count the number of missed, wrong, or mislabeled tags. By counting the overall number of tags (positives) and non-tagged words (negatives), the confusion matrix values are found by subtracting missed tags from negatives (false negatives) and wrong tags from positives (false positives). As we see in Table 6.7, the system performed favorably from human readers' perspectives. Although, it is important to remember that this validation is only considered on the global level of tags being right (tagged correctly or not tagged correctly) or wrong (tagged incorrectly [”wrong” or special “mislabeled” case] or not tagged incorrectly [”missed”]). Because there are multiple classes of potential tag labels, there's actually an accompanying precision and recall score with each type, which we saw with the auto-generated results but lacked sufficient experts to compute individually. Those are encompassed here as “mislabeled” tags, and for the purposes of reporting global results, they have been assumed as tagged incorrectly alongside the “wrong” count. Several performance metrics based on Table 6.7 are listed below.

Table 6.8: Performance metrics for VCD CRF trained for outcomes-oriented labels.

Label	**Precision**	**Recall**	**F1-score**	**Support**
Outcome::Intended_Outcome	1.00	0.50	0.67	10
Outcome::Intended_Outcome_Risk	0.00	0.00	0.00	10
Outcome::Unintended_Outcome	1.00	0.69	0.82	26
Outcome::Unintended_Outcome_Risk	0.91	0.74	0.82	208
Average / Total	0.89	0.70	0.78	254
Unlabeled Text	0.90	0.95	0.93	981

Precision = 0.9007444169

TPR, Recall, Sensitivity = 0.8725961538

TNR, Specificity = 0.9989141058

FPR = 0.001085894234

FNR = 0.1274038462

LR+, Pos Likelihood = 803.5737981

LR-, Neg Likelihood = 0.1275423436

Diagnostic Odds = 6300.44717

Table 6.8 is the other VCD CRF, trained to label outcome-oriented words in the text. In keeping with the findings of Figures 6.6 and 6.7, this CRF uses the same set of features as listed in Table 6.5. *Unintended_Outcome_Risk* is over-represented in the data due to the definition covering complications and side effects from medication use. Side effects are commonly listed in discharge papers, while the actual outcomes remain difficult to find. As with Table 6.6, the "Unlabeled Text" row gives perspective on the CRF's ability to identify relevant text out of large documents.

In Table 6.9, the metrics for the combined CRF running on HCD's clinical notes are listed. *Change_Therapy* and *Stop_Therapy* are still under-represented, but surprisingly, outcome risks weren't found at all by our clinical experts when generating

Table 6.9: Performance metrics for HCD CRF trained to extract both intervention and outcome labels.

Label	**Precision**	**Recall**	**F1-score**	**Support**
General	0.64	0.45	0.53	40
Outcome::Intended_Outcome	0.71	0.37	0.49	100
Outcome::Unintended_Outcome	0.58	0.53	0.55	83
Therapy::Change_Therapy	0.00	0.00	0.00	2
Therapy::Continue_Therapy	0.78	0.51	0.62	74
Therapy::Initiate_Therapy	0.80	0.67	0.73	18
Therapy::Stop_Therapy	1.00	0.38	0.55	8
Average / Total	0.69	0.47	0.55	325
Unlabeled Text	0.94	0.98	0.96	2420

the data. The practitioners for HCD's source do not list side effects in their internal notes with the level of frequency with which they appear in discharge documents. This made finding outcomes themselves slightly easier, but the data set is smaller in general and leaves room for improvement.

The following block gives a short example of the CRF's output text:

> We suspect that Kin has a bleeding GI ulcer. We are lowering his dose of <tag class=“Therapy::Continue_Therapy”>prednisone</tag> for this reason. Please apply topical <tag class=”Therapy::Initiate_Therapy”>cortisone cream</tag> to Kin EM lesions as directed in the medication chart. Kin pancreatitis may be secondary to his suspected GI disease. Recommend rechecking the cPLI on Monday, $2015-07-13$.

This snippet includes redaction from earlier in the pipeline with a substituted patient name and altered dates. Also, the labeled text is offset with tags denoting the recognized type. We see how adept the system is at distinguishing between non-relevant or conversational text and information associated with the intervention and outcome care plan. Notably, the CRF correctly identifies cortisone cream as a new treatment, and it recognizes that prednisnone is not new. This example illustrates the subtlety of the annotation task because prednisnone is technically a changed therapy, which is slightly distinct from a continued one. These minor mistakes partly contribute to the poor performance between classes of labels, even though the CRF is adept at isolating the relevant text.

We are continuing to explore other means of enhancing annotation performance and recognizing outcomes-oriented information in other parts of the patient record. As we have shown, the current stage is able to identify some such materials, albeit only in the free text, but it is more than we were formerly able to achieve from the structured fields in the data.

7. FUTURE WORK

An enormous body of work has been completed thus far, but there are still many avenues for future development. Foremost, the underlying ontology will likely see changes in its next version based on findings from the current applications to veterinary and hospice care data. The pre-processing stage will need to be adjusted accordingly as more domains are added to our data store, including community and public health records. In the future, zoonotic disease may be a worthwhile investigation given the vastness of VCD. Unfortunately, that cannot move forward until comparable human data is available.

The most exciting possibilities lie in data analysis. We have only begun to understand the potential information available in the structured record. Not only is episode grouping going to change from a clustering model to logistic regression in the near future, but more extensive sequential modeling should reveal associations among fields. If there is sufficient data, we would like to form patient cohorts based on a measure of patient similarity. At some point, patient similarity will become a necessary metric for learning typical outcomes and time intervals for a given intervention, and that should carry additional findings.

We are looking forward to bringing more NLP tools to bear. Medical text contains much more information than just outcomes of care, although that has been the focus of our current work. One example is the ability to incorporate diagnostic information from the text with lab results saved in the database. That could augment features of existing automated diagnosing systems. There are also several other algorithms we wish to use in our current pipeline to find ways of improving performance.

8. CONCLUSION

8.1 Research and Competency Questions Revisited

In this paper, we have presented our extensive operations for developing an end-to-end pipeline for moving from raw data sources, through formalization, preprocessing, and analyses, to extracting outcomes-of-care-oriented information from patient records. The primary difficulty associated with this task is the lack of outcomes in most modern EMRs. By using a formal encoding, the Integrated Outcomes Ontology, we were able to phrase exactly what we wished to retrieve from the data in the form of competency questions. Section 3.1 shows the set of seven questions which we have sought to answer, at least partially, throughout the processing and analysis of veterinary and hospice care data. These questions are also related to our central research questions posed in Section 1.2. We will revisit them now in light of the explanation of our work.

1. **Can cross-domain medical records be encoded in a single, feedback-based ontology representing outcomes-oriented concerns?**

   This question ties to all seven IOO competency questions but is best answered by Chapters 3 and 4. We found that, yes, it is possible to create a very simple ontology which can represent the core components of the patient record necessary to label outcomes of care. In our experiments, we use two data sets from different domains, veterinary and hospice care, without adjusting IOO definitions. Only the original mapping must be created to enable encoding; these mappings are given in entirety in Appendix A.

2. **What features are necessary to divide patient records into related segments of episodes of care?**

   This is associated with the second IOO competency question. Episodes of care have been a significant focus of our work, reflected by their presence in our research questions. This is because episodes of care form the bridge between roles/encounters and interventions/outcomes – an outcome must tie back to a related encounter. In our case, using patient age, presenting complaint, diagnosis, and prescribed medications, we can group encounters into related episodes with a rand index score of $83.9 \pm 26.5\%$

3. **Can we extract interventions and, in particular, outcomes data from free medical texts in the absence of a structured record?**

   Interventions and outcomes are connected to IOO's third through seventh competency questions. We had some success identifying interventions in structured records when constraining them to medications. The more flexible solution has been to use CRFs to label likely interventions and outcomes in the patient's medical text. In the veterinary domain, we have been able to identify interventions and outcomes with average F1-scores of 68% and 78%, respectively. In the hospice domain, the cumulative F1-score was a lower 55%, likely due to the smaller set of text in clinical notes and limited amount of written outcomes.

Research questions two and three are not fully answered as we are still pursuing other methods of improving these results. However, we take these early results as encouragement that text mining is a rich resource for augmenting medical information systems without much other patient information.

8.2 Final Remarks

As we continue to apply new methods to this problem, we expect to find better answers to the posed questions. In the meantime, we have been able to build an extensive framework for processing digital patient records to identify outcomes of care. The ultimate goal of this study will be the deployment of a system which can gather outcomes-related information directly from patients, integrate it into the rest of the record in the EMR, and inform future care.

REFERENCES

[1] Richard Hillestad, James Bigelow, Anthony Bower, Federico Girosi, Robin Meili, Richard Scoville, and Roger Taylor. Can electronic medical record systems transform health care? potential health benefits, savings, and costs. *Health affairs*, 24(5):1103–1117, 2005.

[2] Jonathan Pevsner. *Bioinformatics and Functional Genomics.* Wiley Publishing, 2nd edition, 2009.

[3] Kunio Doi. Computer-aided diagnosis in medical imaging: historical review, current status and future potential. *Computerized medical imaging and graphics*, 31(4):198–211, 2007.

[4] Mary K Anthony and Diane C Hudson-Barr. Successful patient discharge: A comprehensive model of facilitators and barriers. *Journal of Nursing Administration*, 28(3):48–55, 1998.

[5] Christopher L Roy, Eric G Poon, Andrew S Karson, Zahra Ladak-Merchant, Robin E Johnson, Saverio M Maviglia, and Tejal K Gandhi. Patient safety concerns arising from test results that return after hospital discharge. *Annals of Internal Medicine*, 143(2):121–128, 2005.

[6] Frederick Baekeland and Lawrence Lundwall. Dropping out of treatment: a critical review. *Psychological bulletin*, 82(5):738, 1975.

[7] John Hsu, Jie Huang, Vicki Fung, Nan Robertson, Holly Jimison, and Richard Frankel. Health information technology and physician-patient interactions: impact of computers on communication during outpatient primary care visits. *Journal of the American Medical Informatics Association*, 12(4):474–480, 2005.

[8] Symone B Detmar, Martin J Muller, Liowina DV Wever, Jan H Schornagel, and Neil K Aaronson. Patient-physician communication during outpatient palliative treatment visits: An observational study. *Jama*, 285(10):1351–1357, 2001.

[9] Christine Sinsky, Lacey Colligan, Ling Li, Mirela Prgomet, Sam Reynolds, Lindsey Goeders, Johanna Westbrook, Michael Tutty, and George Blike. Allocation of physician time in ambulatory practice: A time and motion study in 4 specialtiesallocation of physician time in ambulatory practice. *Annals of Internal Medicine*, 165(11):753–760, 2016.

[10] George A Gellert, Ricardo Ramirez, and S Luke Webster. The rise of the medical scribe industry: implications for the advancement of electronic health records. *JAMA*, 313(13):1315–1316, 2015.

[11] Rebecca T Mercuri. The hipaa-potamus in health care data security. *Communications of the ACM*, 47(7):25–28, 2004.

[12] JC Wyatt. Clinical data systems, part 2: components and techniques. *The Lancet*, 344(8937):1609–1614, 1994.

[13] Wendy W Chapman, Prakash M Nadkarni, Lynette Hirschman, Leonard W D'avolio, Guergana K Savova, and Ozlem Uzuner. Overcoming barriers to nlp for clinical text: the role of shared tasks and the need for additional creative solutions, 2011.

[14] Susan Eggly, Louis Penner, Terrance L Albrecht, Rebecca JW Cline, Tanina Foster, Michael Naughton, Amy Peterson, and John C Ruckdeschel. Discussing bad news in the outpatient oncology clinic: rethinking current communication guidelines. *Journal of Clinical Oncology*, 24(4):716–719, 2006.

[15] Jonathan Mortensen, Matthew Horridge, Mark A Musen, and Natalya Fridman Noy. Applications of ontology design patterns in biomedical ontologies. In *AMIA*, 2012.

[16] Yan Ye, Zhibin Jiang, Xiaodi Diao, Dong Yang, and Gang Du. An ontology-based hierarchical semantic modeling approach to clinical pathway workflows. *Computers in biology and medicine*, 39(8):722–732, 2009.

[17] Cui Tao, Guoqian Jiang, Thomas A Oniki, Robert R Freimuth, Qian Zhu, Deepak Sharma, Jyotishman Pathak, Stanley M Huff, and Christopher G Chute. A semantic-web oriented representation of the clinical element model for secondary use of electronic health records data. *Journal of the American Medical Informatics Association*, 20(3):554–562, 2013.

[18] Anny Kartika Sari, Wenny Rahayu, and Mehul Bhatt. An approach for sub-ontology evolution in a distributed health care enterprise. *Information Systems*, 38(5):727–744, 2013.

[19] Esraa Omran, Albert Bokma, Shereef Abu Al-Maati, and David Nelson. Implementation of a chain ontology based approach in the health care sector. *Journal of Digital Information Management*, 7(5), 2009.

[20] Jim Lumsden, Hazel Hall, and Peter Cruickshank. Ontology definition and construction, and epistemological adequacy for systems interoperability: A practitioner analysis. *Journal of Information Science*, 37(3):246–253, 2011.

[21] Jyotishman Pathak, Harold R Solbrig, James D Buntrock, Thomas M Johnson, and Christopher G Chute. Lexgrid: a framework for representing, storing, and querying biomedical terminologies from simple to sublime. *Journal of the American Medical Informatics Association*, 16(3):305–315, 2009.

[22] Farshad Hakimpour and Andreas Geppert. Resolution of semantic heterogeneity in database schema integration using formal ontologies. *Information Technology and Management*, 6(1):97–122, 2005.

[23] Yugyung Lee, Kaustubh Supekar, and James Geller. Ontology integration: Experience with medical terminologies. *Computers in Biology and Medicine*, 36(7):893–919, 2006.

[24] Vreda Pieterse and Derrick G Kourie. Lists, taxonomies, lattices, thesauri and ontologies: Paving a pathway through a terminological jungle. *Knowledge Organization*, 41(3), 2014.

[25] Markus Strohmaier, Simon Walk, Jan Pöschko, Daniel Lamprecht, Tania Tudorache, Csongor Nyulas, Mark A Musen, and Natalya F Noy. How ontologies are made: Studying the hidden social dynamics behind collaborative ontology engineering projects. *Web Semantics: Science, Services and Agents on the World Wide Web*, 20:18–34, 2013.

[26] Bhaskar Kapoor and Savita Sharma. A comparative study ontology building tools for semantic web applications. *International Journal of Web & Semantic Technology (IJWesT)*, 1(3):1–13, 2010.

[27] David Riaño, Francis Real, Joan Albert López-Vallverdú, Fabio Campana, Sara Ercolani, Patrizia Mecocci, Roberta Annicchiarico, and Carlo Caltagirone. An ontology-based personalization of health-care knowledge to support clinical decisions for chronically ill patients. *Journal of biomedical informatics*, 45(3):429–446, 2012.

[28] Anca Draghici and George Draghici. Cross-disciplinary approach for the risk assessment ontology design. *Information Resources Management Journal (IRMJ)*, 26(1):37–53, 2000.

[29] Nan-Chen Hsieh, Rui-Dong Chiang, and Wen-Pin Hung. Ontology based integration of residential care of the elderly system in long-term care institutions. *Journal of Advances in Information Technology*, 6(3), 2015.

[30] Alan L Rector, Rahil Qamar, and Tom Marley. Binding ontologies and coding systems to electronic health records and messages. *Applied Ontology*, 4(1):51–69, 2009.

[31] Sripriya Rajamani, Elizabeth S Chen, Mari E Akre, Yan Wang, and Genevieve B Melton. Assessing the adequacy of the hl7/loinc document ontology role axis. *Journal of the American Medical Informatics Association*, pages amiajnl–2014, 2014.

[32] Valérie Bertaud-Gounot, Régis Duvauferrier, and Anita Burgun. Ontology and medical diagnosis. *Informatics for Health and Social Care*, 37(2):51–61, 2012.

[33] Peter J Haug, Jeffrey P Ferraro, John Holmen, Xinzi Wu, Kumar Mynam, Matthew Ebert, Nathan Dean, and Jason Jones. An ontology-driven, diagnostic modeling system. *Journal of the American Medical Informatics Association*, 20(e1):e102–e110, 2013.

[34] Mark Hoogendoorn, Peter Szolovits, Leon MG Moons, and Mattijs E Numans. Utilizing uncoded consultation notes from electronic medical records for predictive modeling of colorectal cancer. *Artificial intelligence in medicine*, 69:53–61, 2016.

[35] Jesualdo Tomás Fernández-Breis, José Alberto Maldonado, Mar Marcos, María del Carmen Legaz-García, David Moner, Joaquín Torres-Sospedra, Angel Esteban-Gil, Begoña Martínez-Salvador, and Montserrat Robles. Leveraging electronic healthcare record standards and semantic web technologies for the

identification of patient cohorts. *Journal of the American Medical Informatics Association*, 20(e2):e288–e296, 2013.

[36] Gunjan Mansingh, Kweku-Muata Osei-Bryson, and Han Reichgelt. Using ontologies to facilitate post-processing of association rules by domain experts. *Information Sciences*, 181(3):419–434, 2011.

[37] S Paul, A Kokossis, H Gage, L Storey, R Lawrenson, P Trend, K Walmsley, S Morrison, J Kaye, E Gradwell, et al. A semantically enabled formalism for the knowledge management of parkinson's disease. *Medical informatics and the Internet in medicine*, 31(2):101–120, 2006.

[38] Lori L Popejoy, Mohammed A Khalilia, Mihail Popescu, Colleen Galambos, Vanessa Lyons, Marilyn Rantz, Lanis Hicks, and Frank Stetzer. Quantifying care coordination using natural language processing and domain-specific ontology. *Journal of the American Medical Informatics Association*, pages amiajnl–2014, 2014.

[39] Latanya Sweeney. Replacing personally-identifying information in medical records, the scrub system. In *Proceedings of the AMIA annual fall symposium*, page 333. American Medical Informatics Association, 1996.

[40] Latanya Sweeney. Guaranteeing anonymity when sharing medical data, the datafly system. In *Proceedings of the AMIA Annual Fall Symposium*, page 51. American Medical Informatics Association, 1997.

[41] Sean M Thomas, Burke Mamlin, Gunther Schadow, and Clement McDonald. A successful technique for removing names in pathology reports using an augmented search and replace method. In *Proceedings of the AMIA Symposium*, page 777. American Medical Informatics Association, 2002.

[42] R Miller, JK Boitnott, and GW Moore. Web-based free-text query system for surgical pathology reports with automatic case deidentification. *Arch Pathol Lab Med*, 125:1011, 2001.

[43] Ishna Neamatullah, Margaret M Douglass, H Lehman Li-wei, Andrew Reisner, Mauricio Villarroel, William J Long, Peter Szolovits, George B Moody, Roger G Mark, and Gari D Clifford. Automated de-identification of free-text medical records. *BMC medical informatics and decision making*, 8(1):32, 2008.

[44] Franck Dernoncourt, Ji Young Lee, Ozlem Uzuner, and Peter Szolovits. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, page ocw156, 2016.

[45] Brett R South, Danielle Mowery, Ying Suo, Jianwei Leng, Óscar Ferrández, Stephane M Meystre, and Wendy W Chapman. Evaluating the effects of machine pre-annotation and an interactive annotation interface on manual de-identification of clinical text. *Journal of biomedical informatics*, 50:162–172, 2014.

[46] Iain Hrynaszkiewicz, Melissa L Norton, Andrew J Vickers, and Douglas G Altman. Preparing raw clinical data for publication: guidance for journal editors, authors, and peer reviewers. *Trials*, 11(1):9, 2010.

[47] Brett R South, Shuying Shen, Makoto Jones, Jennifer Garvin, Matthew H Samore, Wendy W Chapman, and Adi V Gundlapalli. Developing a manually annotated clinical document corpus to identify phenotypic information for inflammatory bowel disease. *BMC bioinformatics*, 10(9):S12, 2009.

[48] Benjamin Frederick Ganzfried, Markus Riester, Benjamin Haibe-Kains, Thomas Risch, Svitlana Tyekucheva, Ina Jazic, Xin Victoria Wang, Mahnaz Ahmadifar, Michael James Birrer, Giovanni Parmigiani, et al. curatedovar-

iandata: clinically annotated data for the ovarian cancer transcriptome. *The Journal of Biological Databases and Curation*, 2013.

[49] Özlem Uzuner, Imre Solti, and Eithon Cadag. Extracting medication information from clinical text. *Journal of the American Medical Informatics Association*, 17(5):514–518, 2010.

[50] Serguei V Pakhomov, Anni Coden, and Christopher G Chute. Developing a corpus of clinical notes manually annotated for part-of-speech. *International journal of medical informatics*, 75(6):418–429, 2006.

[51] Lucian Galescu and Nate Blaylock. A corpus of clinical narratives annotated with temporal information. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 715–720. ACM, 2012.

[52] Ozlem Uzuner, Andreea Bodnari, Shuying Shen, Tyler Forbush, John Pestian, and Brett R South. Evaluating the state of the art in coreference resolution for electronic medical records. *Journal of the American Medical Informatics Association*, 19(5):786–791, 2012.

[53] Veronika Vincze, György Szarvas, Richárd Farkas, György Móra, and János Csirik. The bioscope corpus: biomedical texts annotated for uncertainty, negation and their scopes. *BMC bioinformatics*, 9(11):S9, 2008.

[54] Sylvia Siebig, Silvia Kuhls, Michael Imhoff, Julia Langgartner, Michael Reng, Jürgen Schölmerich, Ursula Gather, and Christian E Wrede. Collection of annotated data in a clinical validation study for alarm algorithms in intensive carea methodologic framework. *Journal of critical care*, 25(1):128–135, 2010.

[55] Angus Roberts, Robert Gaizauskas, Mark Hepple, Neil Davis, George Demetriou, Yikun Guo, Jay Subbarao Kola, Ian Roberts, Andrea Setzer,

Archana Tapuria, et al. The clef corpus: semantic annotation of clinical text. In *AMIA Annual Symposium Proceedings*, volume 2007, page 625. American Medical Informatics Association, 2007.

[56] Christine Tsien and James Fackler. An annotated data collection system to support intelligent analysis of intensive care unit data. *Advances in Intelligent Data Analysis Reasoning about Data*, pages 111–121, 1997.

[57] Philipp Bruland, Bernhard Breil, Fleur Fritz, and Martin Dugas. Interoperability in clinical research: from metadata registries to semantically annotated cdisc odm. *Stud Health Technol Inform*, 180:564–8, 2012.

[58] Nicki Tiffin, Janet F Kelso, Alan R Powell, Hong Pan, Vladimir B Bajic, and Winston A Hide. Integration of text-and data-mining using ontologies successfully selects disease gene candidates. *Nucleic acids research*, 33(5):1544–1552, 2005.

[59] Son Doan, Lisa Bastarache, Sergio Klimkowski, Joshua C Denny, and Hua Xu. Integrating existing natural language processing tools for medication extraction from discharge summaries. *Journal of the American Medical Informatics Association*, 17(5):528–531, 2010.

[60] Min Jiang, Yukun Chen, Mei Liu, S Trent Rosenbloom, Subramani Mani, Joshua C Denny, and Hua Xu. A study of machine-learning-based approaches to extract clinical entities and their assertions from discharge summaries. *Journal of the American Medical Informatics Association*, 18(5):601–606, 2011.

[61] Yaoyun Zhang, Buzhou Tang, Min Jiang, Jingqi Wang, and Hua Xu. Domain adaptation for semantic role labeling of clinical text. *Journal of the American Medical Informatics Association*, page ocu048, 2015.

[62] John Lafferty, Andrew McCallum, Fernando Pereira, et al. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289, 2001.

[63] Simon Haykin and Neural Network. Neural networks: A comprehensive foundation. *Neural Networks*, 2(2004):41, 2004.

[64] Larry Reeve et al. Integrating hidden markov models into semantic web annotation platforms. *Technique Report*, 2004.

[65] Narjes Boufaden. An ontology-based semantic tagger for ie system. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 2*, pages 7–14. Association for Computational Linguistics, 2003.

[66] Alexandros Valarakos, Georgios Sigletos, Vangelis Karkaletsis, and Georgios Paliouras. A methodology for semantically annotating a corpus using a domain ontology and machine learning. In *Proceedings of the International Conference in Racent Advances in NLP (RANLP), Borovest, Bulgaria*, 2003.

[67] Lawrence Rabiner and B Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

[68] Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Coffey, Michael Thompson, James Bost, Javier Tejedor-Sojo, and Jimeng Sun. Multi-layer representation learning for medical concepts. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1495–1504. ACM, 2016.

[69] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural net-

works. In *Proceedings of the 1st Machine Learning for Healthcare Conference*, pages 301–318, 2016.

[70] Zuoshuang Xiang, Chris Mungall, Alan Ruttenberg, and Yongqun He. Ontobee: A linked data server and browser for ontology terms. In *ICBO*, 2011.

[71] Amie Lamoreaux Hesbach. Techniques for objective outcome assessment. *Clinical techniques in small animal practice*, 22(4):146–154, 2007.

[72] Hong-Gee Kim. Ontology based adverse event reporting system architecture. In *World library And information congress: 72nd ifla general conference and council*, volume 12, pages 17–20, 2006.

[73] Stephanie J Reisinger, Patrick B Ryan, Donald J O'hara, Gregory E Powell, Jeffery L Painter, Edward N Pattishall, and Jonathan A Morris. Development and evaluation of a common data model enabling active drug safety surveillance using disparate healthcare databases. *Journal of the American Medical Informatics Association*, 17(6):652–662, 2010.

[74] Seth Polsley, Atif Tahir, Muppala Raju, Akintayo Akinleye, and Duane Steward. Role-preserving redaction of medical records to enable ontology-driven processing. In *BioNLP 2017*, pages 194–199, Vancouver, Canada,, August 2017. Association for Computational Linguistics.

[75] Clare Narrod, Jakob Zinsstag, and Marites Tiongco. A one health framework for estimating the economic costs of zoonotic diseases on society. *EcoHealth*, 9(2):150–162, 2012.

[76] Alexis García, James G Fox, Thomas E Besser, et al. Zoonotic enterohemorrhagic escherichia coli: a one health perspective. *ILAR J*, 51(3):221–232, 2010.

[77] Filipe Dantas-Torres, Bruno B Chomel, and Domenico Otranto. Ticks and tick-borne diseases: a one health perspective. *Trends in parasitology*, 28(10):437–446, 2012.

[78] Richard Coker, Jonathan Rushton, Sandra Mounier-Jack, Esron Karimuribo, Pascal Lutumba, Dominic Kambarage, Dirk U Pfeiffer, Katharina Stärk, and Mark Rweyemamu. Towards a conceptual framework to support one-health research for policy on emerging zoonoses. *The Lancet infectious diseases*, 11(4):326–331, 2011.

[79] Jonna AK Mazet, Deana L Clifford, Peter B Coppolillo, Anil B Deolalikar, Jon D Erickson, and Rudovick R Kazwala. A one health approach to address emerging zoonoses: the hali project in tanzania. *PLoS Med*, 6(12):e1000190, 2009.

[80] Stephen Higgs. Zika virus: emergence and emergency, 2016.

[81] Alfonso J Rodriguez-Morales, Antonio Carlos Bandeira, and Carlos Franco-Paredes. The expanding spectrum of modes of transmission of zika virus: a global concern. *Annals of clinical microbiology and antimicrobials*, 15(1):13, 2016.

[82] Massimo Franchini and Claudio Velati. Blood safety and zoonotic emerging pathogens: now its the turn of zika virus! *Blood Transfusion*, 14(2):93, 2016.

[83] Michael J Day. One health: the importance of companion animal vector-borne diseases. *Parasites & vectors*, 4(1):49, 2011.

[84] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

[85] Kent A Spackman, Keith E Campbell, and Roger A Côté. Snomed rt: a reference terminology for health care. In *Proceedings of the AMIA annual fall symposium*, page 640. American Medical Informatics Association, 1997.

[86] George H Dunteman. Principal component analysis (quantitative applications in the social sciences), 1989.

[87] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[88] Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

[89] Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu. Using of jaccard coefficient for keywords similarity. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1, page 6, 2013.

[90] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[91] Douglas Steinley. Properties of the hubert-arable adjusted rand index. *Psychological methods*, 9(3):386, 2004.

[92] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[93] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, 1(1-4):43–52, 2010.

[94] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, 2003.

[95] Akiko Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65, 2003.

[96] Wen Zhang, Taketoshi Yoshida, and Xijin Tang. A comparative study of tf* idf, lsi and multi-words for text classification. *Expert Systems with Applications*, 38(3):2758–2765, 2011.

[97] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Numerische mathematik*, 14(5):403–420, 1970.

[98] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[99] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, volume 13, pages 746–751, 2013.

[100] Todd K Moon. The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60, 1996.

[101] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, pages 281–297. Oakland, CA, USA., 1967.

[102] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[103] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[104] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.

[105] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics, 2010.

[106] Percy Liang. *Semi-supervised learning for natural language*. PhD thesis, Massachusetts Institute of Technology, 2005.

[107] Leon Derczynski, Sean Chester, and Kenneth S Bøgh. Tune your brown clustering, please. In *International Conference Recent Advances in Natural Language Processing, RANLP*, volume 2015, pages 110–117. Association for Computational Linguistics, 2015.

[108] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[109] Matthijs J Warrens. On the equivalence of cohens kappa and the hubert-arabie adjusted rand index. *Journal of classification*, 25(2):177–183, 2008.

[110] Kenji Hirohata, Naoaki Okazaki, Sophia Ananiadou, Mitsuru Ishizuka, and Manchester Interdisciplinary Biocentre. Identifying sections in scientific abstracts using conditional random fields. In *IJCNLP*, pages 381–388, 2008.

[111] Yuta Tsuboi, Yuya Unno, Hisashi Kashima, and Naoaki Okazaki. Fast newton-cg method for batch learning of conditional random fields. In *AAAI*, 2011.

[112] Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics, 2006.

[113] Martin F Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

APPENDIX A

DATA PROCESSING AND REDACTION MAPPINGS

The next few tables provide reference information showing the exact mapping between the original data sources and the cleaned, redacted data. Table A.1 reviews the old storage location of each field and the new, cleaned data location alongside the intention of that mapping for VCD. The intention column is primarily influenced by compliance with the ontology.

HCD includes several tables. Table A.2 is the primary mapping between original and new fields. Because HBV is not reliant on regular discharges, being a hospice setting, the only free text comes in the form of clinical notes. Table A.3 shows the supplemental mappings for notes and problems. Also, as HBV supports limited measurement of goals and outcomes, additional tables were used as reference for the resulting data set. Table A.4 shows these references and their purpose.

Table A.1: VCD mappings from the original VMIS data.

Old Table	Old Field	Intention	New Table.Field
mr_entry	patient_id	Gather patient info for ontology	Not stored, only used for later calls
vw_case_card (view)	patient_name	Used to generate pseudonyms for role-based redaction of known names	protocol_patients. patient_name (pseudonym only)
	owner_fname		Not stored, only for role-based redaction
	owner_lname		
admission	admission_datetime	Dates collected to preserve timeframes	protocol_encounters. start_time (offset)
	discharge_datetime		protocol_encounters. end_time (offset)
	attending_clinician_id	Used to locate clinician	Not stored, only used for later calls
	presenting_complaint	Reason for visit which informs outcome from previous visit	protocol_encounter. reason (redacted)
	id	Used to locate mr_entries	Not stored, used only for later calls
vmis_user	fname	Clinician name for role-based redaction	Not stored, only for role-based redaction
	lname		
mr_entry	class	Used to group type	protocol_interventions. class
	date_recorded	Used to allow in-text date offset to reflect appropriate year	protocol_interventions. recorded_date (offset)
	description	Describes MR type or points to document	Protocol_interventions. description (redacted)
	id	Used to reference full documents	protocol_interventions. mr_entry
document	rtf_document	Full text MRs retrieved and processed for more information	protocol_interventions. fulltext (redacted) protocol_interventions. followup (redacted)
	image_document	Full text of binary files where RTF is not used	protcol_interventions. full_text (redacted)
medication	directions	Medication directions entry for more detail about medical entry	protocol_interventions. full_text (redacted) where applicable
necropsy	clinical_history	More details about patient history	protcol_interventions. full_text (redacted) where applicable

Table A.2: HCD mappings from the original HBV data.

Old Table	**Old Field**	**Intention**	**New Table.Field**
PT_BASIC	patient_id	Patient info	Not stored
	name_full	For redaction	protocols_patient.patient_name (pseudonym only)
	notes	Patient notes	protocol_patients.notes (redacted)
PT_ADMISSION	patient_id	Used to reference other tables	protocol_admission.patient_id (new id)
	admit_date	Dates collected to preserve timeframes	protocol_admission.admit_date (offset)
	termination_date		protocol_admission.termination_date (offset)
	date_of_birth	Ignore age >= 89	protocol_admission.patient_age
	RES_ROLE_TABLE	Clinician name collected for role-based redaction	Not stored
	RES_ROLE_DESCRIPTION		
	Caregiver_code	Caregiver type	protocol_admission.caregiver_code
	latest_class	Whether patient is palliative or hospice	protocol_admission.latest_class
PTC_INTERVENTIONS	start_date	Dates collected to preserve time frames by using same offset for a patient	protocol_encounter_interventions.start_date (offset)
	end_date		protocol_encounter_interventions.end_date(offset)
	problem_code	For storing the problem warranting intervention	protocol_encounter_interevention.problem_code
	intervention_code	For storing interevention_code	protocol_encounter_intervention.intervention.code
C_Intervention	I_Description	For the corresponding intervention description	protocol_encounter_intervention.description

Table A.3: HCD mappings for supplemental table data from HBV.

PTC_Clinical_notes	note_id	Auto-generated identifier	protocol_notes. note_id (new id)
	patient_id	To maintain relationships	protocol_notes_ patient _id (new id)
	discipline_code	Discipline for providers code	Protocol_notes. discipline_code
	cn_text	Clinical notes	Protocol_interventions .cn_text (redacted)
	create_date	Preserving time frame for notes	protocol_notes. create_date (offset)
PTC_PROBLEM	patient_id	Reference the patient_problem	protocol_problems. patient_id
	ptc_problem_id	To track problem with other tables	protocol_problems. problem_id
	problem_code	Code for the problem	protocol_problems. problem_code
C_Problem	problem_ description	Code for the problem	protocol_problems. description (passing)

Table A.4: HCD support information referenced from HBV tables.

Old Table	Old Field	Redaction Strategy
PTC_DIAGNOSIS	diagnosis_id	To get the diagnosis description from C_Diagnosis
	patient_id	Self generated random patient id to reference the diagnosis for single pass for redaction
	diagnosis_date	Random offset to change date
C_DIAGNOSIS	diagnosis	To get the generic diagnosis description associated with the diagnosis_id
PTC_INTERVENTION_VARIANCE	PP_INTERVENTION_ID	To reference the patient intervention from PTC_INTERVENTION
	Variance	Variance_code to get generic description of variance from C_Interevention_variance
	create_date	Date will be offset
C_Intervention_Variance	description	To get the generic variance description
PTC_GOAL	pp_goal_id	For referencing the goal id
	patient_id	Self generated random patient id to reference the patient goal
	goal_code	Code for the goal
C_Goal	goal_description	To access the generic description
PTC_GOAL_VARIANCE	PP_GOAL_ID	Goal id referenced from prc_goal
	MEETS	Boolean value to assess the goal
	Variance	Code for goal variance
C_Goal_Variance	Description	Generic goal description

APPENDIX B

WORD CLOUDS FOR DISCHARGE SUMMARY SEGMENTS

These word clouds, generated with wordclouds.com[1], show the weighted relationships among the 60 most frequent words in each segment for treatment, exercise, and diet.

[1] http://www.wordclouds.com/

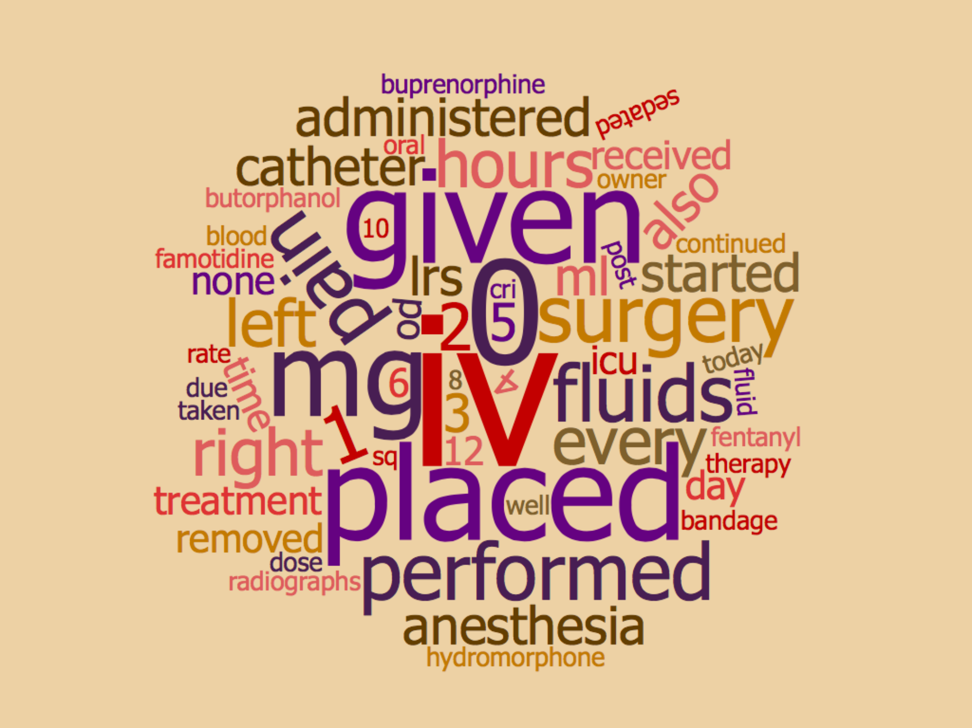

Figure B.1: Word cloud visualizing the top 60 words in the "treatment" subsection.

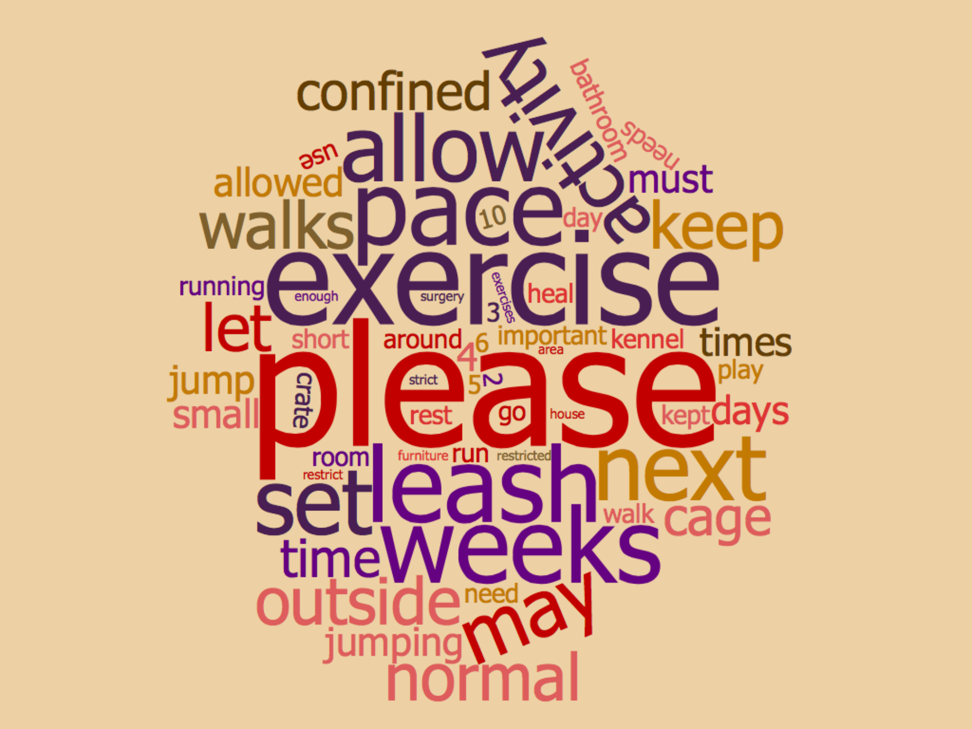

Figure B.2: Word cloud visualizing the top 60 words in the "exercise" subsection.

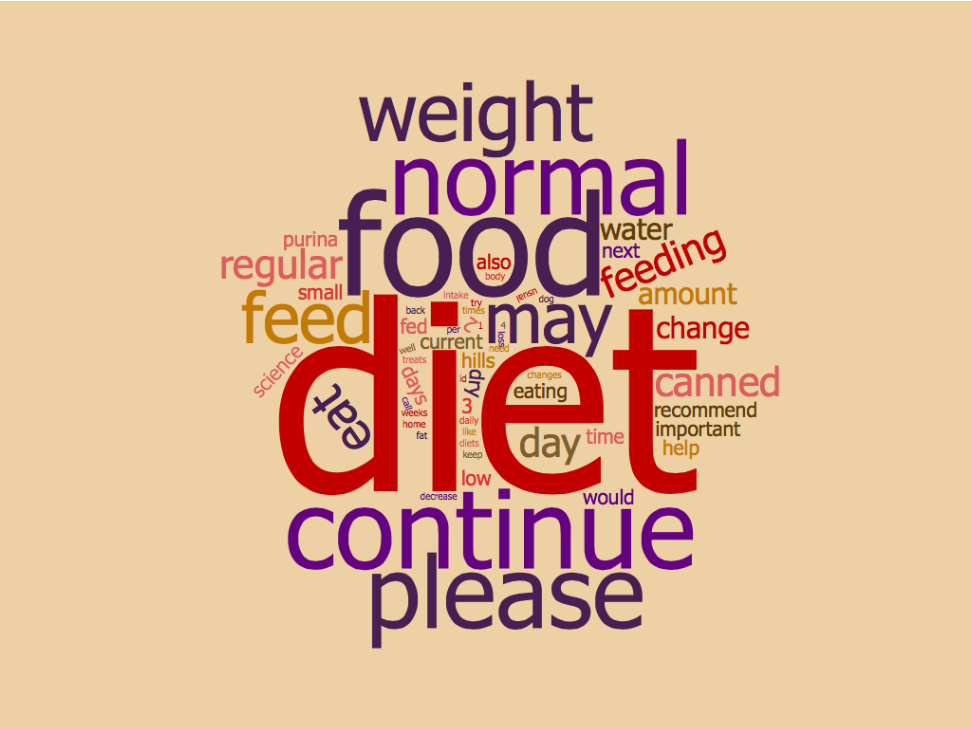

Figure B.3: Word cloud visualizing the top 60 words in the "diet" subsection.