A CLINICAL DECISION SUPPORT SYSTEM FOR THE IDENTIFICATION OF POTENTIAL HOSPITAL READMISSION PATIENTS

by

Christopher Baechle

A Dissertation Submitted to the Faculty of

The College of Engineering and Computer Science

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

Florida Atlantic University

Boca Raton, FL

August 2017

A CLINICAL DECISION SUPPORT SYSTEM FOR THE IDENTIFICATION OF POTENTIAL HOSPITAL READMISSION PATIENTS

by

Christopher Baechle

This dissertation was prepared under the direction of the candidate's dissertation advisor, Dr. Ankur Agarwal, Department of Computer and Electrical Engineering and Computer Science, and has been approved by the members of his supervisory committee. It was submitted to the faculty of the College of Engineering and Computer Science and was accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

SUPERVISORY COMMITTEE:

Ankur Agarwal, Ph.D.
Dissertation Advisor

Ravi Behara, Ph.D.

Hari Kalva, Ph.D.

Xingquan Zhu, Ph.D.

Nurgun Erdol, Ph.D.
Chair, Department of Computer and Electrical Engineering and Computer Science

Mohammad Ilyas, Ph.D.
Dean, College of Engineering and Computer Science

Deborah L. Floyd, Ed.D.
Dean, Graduate College

June 28, 2017
Date

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Ankur Agarwal, for his guidance throughout my education and research. Dr. Agarwal's wisdom and encouragement has kept me on the path to success. Without his support, this dissertation would not have been possible. I am thankful to Dr. Xingquan Zhu for his help in ensuring proper methodology and correctness of my research, as well as the immense technical knowledge I gained from his courses. I would also like to thank Dr. Ravi Behara for providing insight into the clinical use of this research. I am thankful to Dr. Hari Kalva for providing invaluable feedback of this work.

I am thankful to Dr. Robert Cooper, Dr. Martin Solomon, Dr. Dingding Wang, Dr. Thomas Fernandez, Dr. Taghi Khoshgoftaar, Dr. Lofton Bullard, Dr. Oge Marques, Dr. Georgiana Carvalho, and Dr. Ravi Shankar for providing me the computer science education that helped shape my future. I would also like to thank Jean Mangiaracina for her help in navigating the graduate process and procedures.

I would like to thank Florida Atlantic University which has given me the platform to reach my full potential.

Finally, special thanks to Alison Lukowsky and my family for their patience and support.

ABSTRACT

Author:	Christopher Baechle
Title:	A Clinical Decision Support System for the Identification of Potential Hospital Readmission Patients
Institution:	Florida Atlantic University
Dissertation Advisor:	Dr. Ankur Agarwal
Degree:	Doctor of Philosophy
Year:	2017

Recent federal legislation has incentivized hospitals to focus on quality of patient care. A primary metric of care quality is patient readmissions. Many methods exist to statistically identify patients most likely to require hospital readmission. Correct identification of high-risk patients allows hospitals to intelligently utilize limited resources in mitigating hospital readmissions. However, these methods have seen little practical adoption in the clinical setting. This research attempts to identify the many open research questions that have impeded widespread adoption of predictive hospital readmission systems.

Current systems often rely on structured data extracted from health records systems. This data can be expensive and time consuming to extract. Unstructured clinical notes are agnostic to the underlying records system and would decouple the predictive analytics system from the underlying records system. However, additional concerns in clinical natural language processing must be addressed before such a system can be implemented.

Current systems often perform poorly using standard statistical measures. Misclassification cost of patient readmissions has yet to be addressed and there currently exists a gap between current readmission system evaluation metrics and those most appropriate in the clinical setting. Additionally, data availability for localized model creation has yet to be addressed by the research community. Large research hospitals may have sufficient data to build models, but many others do not. Simply combining data from many hospitals often results in a model which performs worse than using data from a single hospital.

Current systems often produce a binary readmission classification. However, patients are often readmitted for differing reasons than index admission. There exists little research into predicting primary cause of readmission. Furthermore, co-occurring evidence discovery of clinical terms with primary diagnosis has seen only simplistic methods applied.

This research addresses these concerns to increase adoption of predictive hospital readmission systems.

A CLINICAL DECISION SUPPORT SYSTEM FOR THE IDENTIFICATION OF POTENTIAL HOSPITAL READMISSION PATIENTS

TABLES

FIGURES

CHAPTER 1: INTRODUCTION

1.1 Introduction

Prediction of unplanned all-cause 30-day hospital readmissions has become increasingly important in the past decade. Rising healthcare costs (shown in Figure 1.1) have caused hospitals to shift focus from quantity of care to quality of care. As a consequence, many hospitals have begun to reanalyze patient care. A primary metric of care quality is unplanned hospital readmission [1]. Patients visiting the hospital should not only be treated for immediate symptoms but helped with the necessary steps to prevent unnecessary future hospital visits. Ideally, hospitals could provide every patient with a home healthcare professional to prevent readmission. Patients under constant watch by a home healthcare professional would be unlikely to require hospital visits. In practice, this would be a poor usage of resources and prohibitively expensive. Predictive analytics offers the potential for hospitals to predict patients most likely to need readmission. However, current predictive models have many shortcomings. Many systems are created from clinical experience and lack statistical foundation. Systems built from statistical learning approaches often use general purpose techniques with little concern for the properties of clinical data. This has led to poor model performance despite a large body of research.

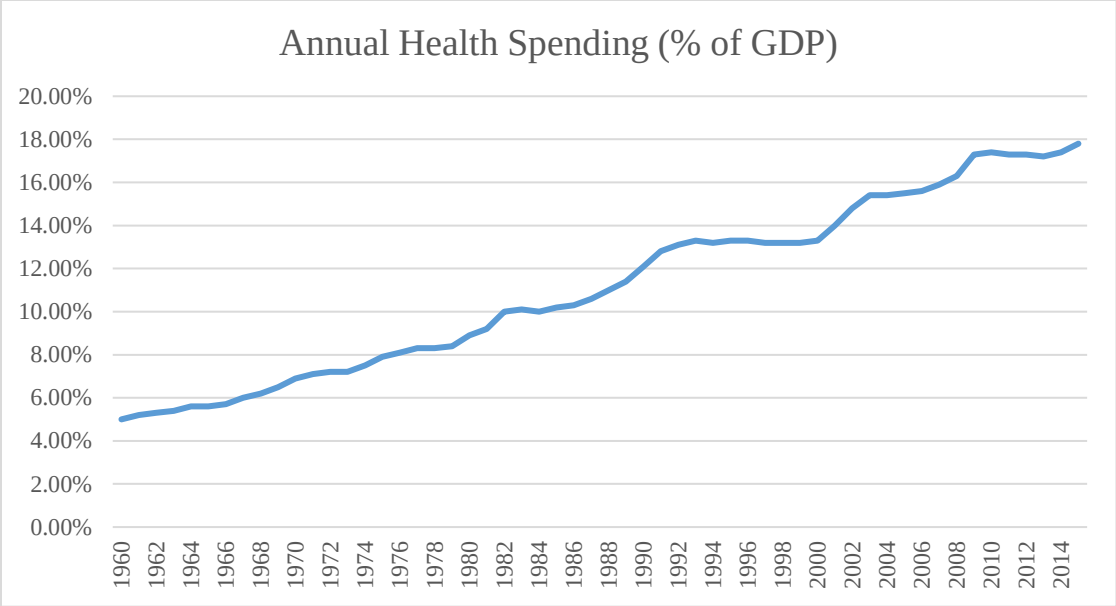

Figure 1.1 Annual Health Spending as a Percentage of GDP, 1960 – 2015 [2]

Currently, hospitals often attempt to lower readmission rates using low cost methods which prevent the largest number of readmissions (often termed “low hanging fruit”). Many times these do not require data analytics and may be as simple as allowing patients to use the hospital pharmacy to fill prescriptions. However, a subset of patients still require targeted care after broad approaches are applied. In order to facilitate the practical adoption of Predictive Hospital Readmission Systems (PHRS), ease of implementation, creation of appropriate performance metrics, and improved performance must be addressed.

1.2 Motivation

The past several years have seen a shift in the focus of hospital priorities. Historically, medical professionals at hospitals were expected to treat patients for immediate symptoms and short term care. Patients were often expected to follow-up with visits to primary care or specialist facilities on their own. In practice, many patients often did not follow-up and would require re-hospitalization several weeks later. This problem

is most pronounced in chronic disease patients. Chronic disease patients would often use hospital facilities as a surrogate for primary care. The Emergency Medical Treatment and Active Labor Act (EMTALA) passed in 1986 stipulated that hospitals accepting federal money though Medicare must treat patients regardless of ability to pay [3]. Patients unable to receive treatment through primary care facilities not subject to EMTALA began to use hospitals to manage chronic conditions. Additionally, patients receiving Medicare benefits would use hospitals to manage chronic conditions. Medicare reimbursed hospitals for each visit in isolation and hospitals had little financial incentive to assist patients with long term management of chronic diseases.

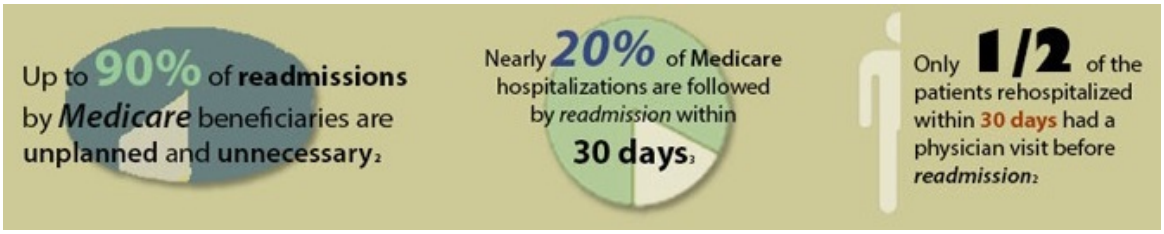

Figure 1.2 Infographic of Readmission Statistics (Source: www.hin.com/infographics)

The Robert Wood Johnson Foundation studied this phenomenon, known as the "revolving door" and found that one in six Medicare patients were readmitted to a hospital within thirty days of their original visit [4]. As shown in Figure 1.2, many hospital readmissions are unnecessary. Additionally, the report found that hospitals were not making progress toward lowering hospital readmissions. Patients were not less likely to require readmission in 2010 than in 2008. There existed little motivation by hospitals and patients to remedy the issue. Payers began to take notice however as receiving routine care in a hospital setting is often significantly more expensive than the same treatments in a primary care or specialist setting.

The passage of the Hospital Readmissions Reduction Program (HRRP) caused hospitals to reanalyze treatment [5]. The HRRP stipulates that the Centers for Medicare and Medicaid Services (CMS) provides hospitals accepting Medicare target rates for hospital readmission. These rates are based on patient demographics and disease distribution. Hospitals found to have excess readmissions receive financial penalties. In some instances, a single patient readmission may exceed the potential Medicare reimbursement by many times [6]. Long-term quality of care quickly became a priority for many hospitals due to these large financial penalties.

Many strategies exist which attempt to reduce unplanned hospital readmissions. One potential strategy is to provide patient education and basic follow-up. This method can be applied equally to all patients and many programs using this strategy currently exist. For example, research at one hospital found patients would often not fill prescription medications prescribed during their visit [4]. To address this problem, patients are now encouraged to have their prescriptions filled directly at the hospital pharmacy. Researchers found this method to drastically increase the number of patients which had their prescriptions filled. Although these methods have been found to be effective, Figure 1.3 and Figure 1.4 show penalties to be increasing.

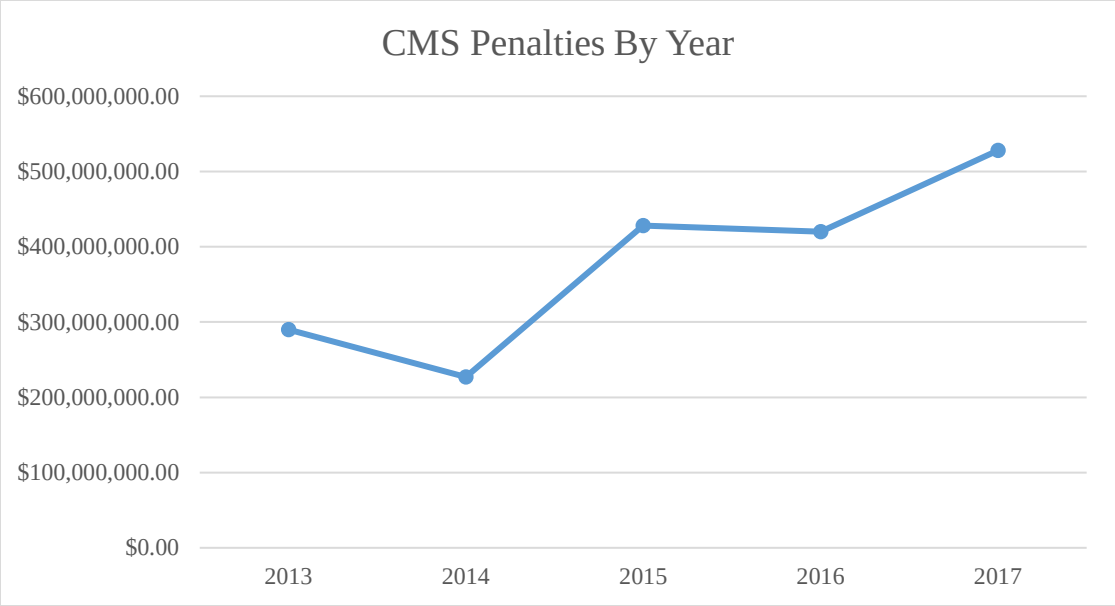

Figure 1.3 Estimated CMS Penalties by Year [7]

Although many hospital readmission predictive models exist, several shortcomings have slowed practical adoption. Current systems are often difficult to implement. Many Electronic Health Record (EHR) systems are available and extracting data in the necessary formats for each model may be time consuming and expensive. Few systems have used physician's notes as the primary data source in readmission prediction. These notes are often referred to as unstructured text and require additional research for practical use. However, from the prospective of the medical facility, unstructured physician's notes are often relatively easy to extract from patient records systems. There exists far less research in the area of PHRS using unstructured text. However, the potential benefits for practical adoption are considerable.

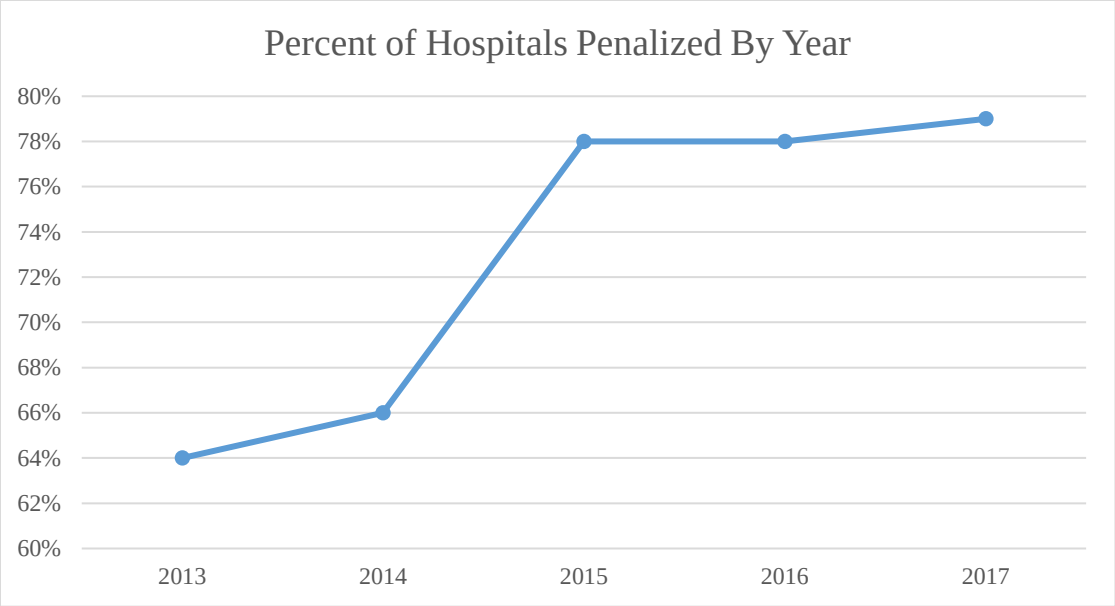

Figure 1.4 Percent of Hospitals Receiving CMS Penalties Nationally [7]

Furthermore, there has been little progress in combining data from differing hospitals. Because many journals do not publish negative results, there is little data regarding the number of researchers that have attempted this and failed. Although some researchers may have a potentially large number of instances available from many hospitals, inclusion of this data often results in models performing worse than only including a targeted subset. This data contains potentially useful information. Models which can incorporate dissimilar hospitals intelligently could potentially increase their performance.

Cost has received little attention in hospital readmission models. Hospitals are often interested in cost minimization but researchers largely report results in terms of common statistical measures such as accuracy and c-statistic. Additionally, current models ignore misclassification cost. The cost of a False Positive (FP) is rarely the same as a False Negative (FN) in both financial and abstract terms.

Many additional smaller but important concerns exist which are not currently being

addressed by the hospital readmissions research community. Current systems often only present binary classification results that can be difficult to interpret. However, many medical professionals prefer Clinical Decision Support Systems (CDSS) that present evidence and probabilities of readmission. Systems are often difficult to compare and primitive when compared to other predictive modeling research domains such as social media and retail. General purpose algorithms and approaches have been applied to this domain which has so far proven ineffective. Due to financial, ethical, and resource motivations alongside the current limited performance and adoption rates it is clear this field of research has many open problems which need to be addressed in a timely manner.

1.3 Contributions

The main contributions of this work include the following:

- The design and implementation of a PHRS using discharge summaries. Using so-called unstructured text often results in simpler and potentially cheaper readmission systems as discharge summaries are easy to extract from patient records systems.
- The creation of a new performance metric based directly on Medicare readmission penalties. This metric more accurately reflects the motivations many hospitals have for interest in PHRS than currently used metrics.
- The design and implementation of a framework for selecting the optimal number of patients to minimize cost.
- The design and implementation of a topic modeling framework to allow combination of differing hospital patient records in a single model. This system is among the first to successfully combine differing hospital data while improving cost. Current systems often discard data from hospitals which contain differing

patient demographics as simply combining datasets results in poor performance.

- Proposed methodology and benchmark of a PHRS using a recently released federal readmissions dataset. Current PHRS are often difficult or impossible to reproduce due to data privacy concerns. Systems are rarely directly compared and the release of this dataset allows comparison of such systems. This work is among the first to create a baseline methodology for feature engineering and feature selection for this dataset. Few publications have been released using this dataset for predictive hospital readmission and are difficult to compare due to differing methodologies.
- The design and implementation of a PHRS predicting primary cause of readmission. Current systems often only attempt to predict readmission as a binary feature. The prediction of primary cause allows medical professionals to more intelligently attempt to mitigate individual readmissions.
- The design and implementation of an improved methodology for finding co-occurring evidence of diseases, symptoms, and medications. Current systems often use co-morbidity indexes such as the Charlson index which are often built purely based on correlations. Finding co-occurring terms with potentially similar root causes may allow for the creation of more intelligent co-morbidity indexes. Additionally, a big data approach is introduced to allow national and international clinical datasets to be used.

1.4 Organization

Chapter 2 presents related works of historical and current PHRS. Several types of PHRS are available, including score based and statistical learning based. Innovations and shortcomings of each system are evaluated. Chapter 3 presents the necessary background

for this research. NLP is heavily employed in many parts of this research. General purpose NLP approaches and tools are first analyzed, followed by NLP approaches focused on the clinical and medical domain. Feature selection is commonly used to find the most important features of a dataset and explored in this chapter as well. Finally, classification and evaluation techniques are reviewed.

Chapter 4 introduces the methodology for this research. Clinical NLP for predicting hospital readmission serves as the foundation. Cost estimation techniques and the introduction of cost a new evaluation metric are then presented. Methods for combining hospital data using latent topic models are outlined and integrated with cost approaches. Predicting primary cause of readmission utilizing previously proposed methods is introduced. Finally, a method for finding co-occurring evidence discovery of clinical terms is presented.

Chapter 5 presents the experimental results of proposed methodologies. These results include traditional evaluation metrics such as AUC, as well as proposed evaluation metrics such as cost. The effectiveness of combining hospital data vs existing methods is then analyzed. Predicting primary cause of readmission results are presented as well as effectiveness of clinical term evidence discovery. Chapter 6 concludes this work and summarizes results and conclusions. Additionally, future directions of this research are discussed.

CHAPTER 2: RELATED WORKS

2.1 Early Systems

The cost of hospital readmissions have been studied since the 1970's. Researchers at that time determined that up to 22% of discharged patients were readmitted to the hospital within 60 days costing approximately $2.5 billion annually (over $12 billion adjusted for inflation in 2017) [8]. The earliest known published PHRS was created in the early 1980's [9]. Researchers created a system with 20 independent variables using 420,894 discharge instances from 270,266 Medicare beneficiaries. Patients readmitted within 60 days of discharge were considered a readmission. A Chi-Square (CS) test was performed to evaluate the importance of each independent variable upon the dependent variable and a p-value of 0.01 used to signify statistical significance.

Table 2.1 shows the mean value for readmission vs non-readmission instances. The number of recent discharges, chronic diseases, and surgeries performed were found by the authors to be among the most significant variables. A multivariate regression model was created and validated using 10,522 unseen instances. A probability of readmission was assigned to each test instance and validated against the ground truth. The system was found to have statistically significant predictive power. However, statistical measures such as c-statistic were not reported and comparison to modern systems difficult.

Variable	**Readmitted**	**Not Readmitted**	**p-value**
Age (years)	72.3	73.2	<.0001
Sex (% male)	50.0	44.5	<.0001
Race (% white)	10.3	10.0	.57
Disability status (% disabled)	11.3	8.9	<.0001

Supplemental Medicaid coverage (%)	15.8	12.8	<.0001
Lives in Northeast (%)	20.1	22.4	<.005
Lives in North Central (%)	28.1	29.2	.14
Lives in West (%)	16.4	14.4	<.001
Discharges in 60 days prior	.45	.23	<.0001
Discharges with same diagnosis	.14	.05	<.0001
Surgery performed (%)	25.4	32.6	<.0001
Length of stay (mean days per case)	14.1	13.1	.27
Hospital reimbursement	1,450.5	1,423.3	.35
Admission for nonchronic disease (%)	14.7	20.7	<.0001
Hospital in urban area (%)	67.2	71.2	<.0001
Community hospital (%)	97.8	97.9	.67
State or local hospital (%)	22.9	20.0	<.0001
For-profit hospital (%)	5.7	5.4	.41
Hospital with <100 beds (%)	19.9	16.4	<.0001
Teaching hospital (%)	15.8	16.0	.83

Table 2.1 Variables Selected for Early PHRS

Other early work from the 1980's includes a readmission system from researchers at Harvard [10]. Unlike the work of Anderson et al., this system included independent variables containing disease diagnosis in addition to patient demographics. A much smaller patient cohort was studied using 4,769 patients from Beth Israel Hospital. Table 2.2 shows the disease diagnosis with highest risk for readmission. Validation of the model used similar methodology as Anderson et al. and only statistical significance was published as a metric to validate the model's predictive ability.

Diagnosis	**Risk for Readmission, 95% C.I.**
AIDS	3.3 (1.4-7.8)
Asthma	0.9 (0.7- 1.4)
Cancer	2.8 (2.4- 3.4)
Cerebrovascular disease	1.0 (0.8- 1.3)
Chronic lung disease	1.4 (0.8- 2.3)
Diabetes mellitus	1.4 (1.2-1.7)
Gastrointestinal bleeding	1.1 (0.8-1.5)
Disease of esophagus, stomach, liver, biliary tract, pancreas, and bowel	0.8 (0.6-1.0)
Ischemic heart disease	1.0 (0.9- 1,2)
Heart failure and cardiomyopathy	1.8 (1.5-2,1)
Psychiatric illness	1.2 (0.9- 1.6)

Renal failure, nephritis	2.3 (1.7- 3.0)

Table 2.2 Variables Selected for Disease-based PHRS

2.2 Score-Based Systems

The earliest identifiable work using c-statistic (also known as AUC) as the primary measure of performance is from the late 1990's by Philbin et al. [11]. This work concentrates on a patient sub-population; that of Congestive Heart Failure (CHF) patients. The patient cohort for this research was 42,731 patients in New York state in the year 1995. This work differs from previous systems in that it chose a sub-population of available patients rather than analyze all available instances. Additionally, this system used the presence of several co-morbidities as features which were not present in previous reviewed works. This is primarily due to the selection of a patient sub-population. Many other features are similar to previous works.

This model was an early attempt at creating a so-called risk score. Previous systems used regression models derived from training data and made no mention of applying the model without modification to differing patient datasets. However, this system identified the most important variables using the CS test. After identifying the ranking of variables, a point value is assigned to each variable. For example, if the patient is receiving Medicare benefits, a point is added to their score. However, if the patient is discharged to a skilled nursing facility, a point is subtracted. This point system is shown in Table 2.3. Binary classification for each possible score boundary was used and the maximum c-statistic was found using +11 as the boundary. Patients with a score +11 or greater were considered readmissions while those below +11 were considered non-readmissions. Using this criteria, a c-statistic of 0.60 was obtained. This is generally considered to have a poor discriminative ability. A c-statistic of 0.50 represents random classification and 1.0 representing perfect

classification.

Feature	Score
Baseline value	+4
Black race	+1
Medicare insurance	+1
Medicaid insurance	+1
Home health care services after discharge	+1
Ischemic heart disease	+1
Valvular heart disease	+1
Diabetes mellitus	+1
Renal disease	+1
Chronic lung disease	+1
Idiopathic cardiomyopathy	+1
Prior cardiac surgery	+1
Use of telemetry during index hospitalization	+1
Treatment at a rural hospital	-1
Discharge to a skilled nursing facility	-1
Echocardiogram performed during index hospitalization	-1
Cardiac catheterization performed during index hospitalization	-1
Range of possible scores	0 to 15

Table 2.3 Score-based PHRS

Research by Van Walraven et al. represents recent research of score-based systems. This system, known as LACE uses only four variables for the calculation of a readmission score. A cohort of 4,812 patients from 11 Ontario hospitals were analyzed and 48 variables extracted from the data source. Regression models were created and variables found to be most important to the model were included. These were found to be length of stay, acuity of admission, Charlson comorbidity index score, and number of hospital visits in the previous six months, forming the acronym LACE. The scoring criteria for LACE is shown in Table 2.4.

Attribute	Value	Points
Length of stay (L)	< 1	0
	1	1
	2	2
	3	3
	4-6	4

	7-13	5
	≥ 14	7
Acute admission (A)	Yes	3
Comorbidity score (C)	0	0
	1	1
	2	2
	3	3
	≥ 4	5
Emergency room visits in previous 6 months (E)	0	0
	1	1
	2	2
	3	3
	≥ 4	4

Table 2.4 Scoring System for LACE

This system achieved a c-statistic of 0.693 for the validation dataset. This is considerably better than previous score-based work. LACE became very popular as it is very easy to implement. Since only four variables are used, direct integration with an EHR system is not necessary. A four question calculator can be easily implemented as a separate software system or simply hand calculated by medical staff. Due to its popularity, many PHRS use LACE as a benchmark for which to compare proposed systems and new datasets [12]–[16]. Although LACE showed improvements in c-statistic over previous score-based systems, the predictive power of LACE on additional datasets varied greatly and was as low as 0.55 [12].

This underscores a fundamental deficiency of score-based systems. These systems often perform well on the data sources from which they were built, but perform poorly when applied to hospitals with differing patient demographics and distribution of features. Scoring systems devised by researchers at Yale have additionally acknowledged low c-statistic when validating against additional datasets despite being endorsed by Hospital-to-Home National Quality Initiative and the Veterans Affairs Quality Enhancement Research

Initiative [17]. Score-based systems are popular due to their simplicity and transparency but there exists little evidence they are effective at predicting patient readmissions in practice.

2.3 Modern Systems

A 2011 review by Kansagara et al. summarizes the work of many modern PHRS [18]. The review looked at 26 different systems using the c-statistic as the primary performance metric. Scores ranged from 0.55 to 0.83 [19], [20]. Although many systems were reviewed, direct comparison of systems is difficult as data sources vary greatly. The largest dataset analyzed contained over 1.4 million patients from England's National Health Service (NHS) [21]. The smallest dataset contained only 487 patients [22]. Many of the reviewed works used regression models for classification while others used scoring systems. Little consensus is reached however regarding best approaches and many systems are very similar to early works nearly 30 years old. New approaches often introduce new variables or use a differing patient sub-population. A summary of modern systems is outlined in Table 2.5.

PHRS	Size	Variables	Portability	Data Source	Algorithm	Evaluation Metric
Philbin et al.	42,731 patients	Demographics, co-morbidities, and hospital.	Any hospital without rebuilding model.	1995 hospital discharges for New York State. CHF patients.	Logistic regression to discover variables.	c-statistic
Yu et al.	2,441, 26,520, and 45,785 patients	Clinical, lab, and demographic	Methodology to rebuild per-disease	Medicare and Medicaid eligible patients	SVM and Cox regression for readmissi	c-statistic

	.		and per-hospital	with CHF, AMI, or Pneumonia.	on prediction.	
Au et al.	59,652 patients	Comorbidity, LACE, and demographic	Any hospital without rebuilding model	Alberta hospital from 1999 to 2009 for HF patients.	Random forest and GINI to discover variabels	c-statistic
Sushmita et al.	221,000 instances	Billing, administrative, clinical, labs	Methodology to rebuild per-disease and per-hospital	General patient population. "All cause" readmission.	SVM, LR, RF, CART, and GBM for readmission prediction	Sensitivity, specificity, precision, and accuracy
Hummel et al.	1,807 patients	Administrative and clinical	Any hospital without rebuilding model	Medicare HF inpatients from 2002 to 2004.	Logistic regression to discover variables	c-statistic
Hosseinzadeh et al.	619,274 instances	Labs, drugs, demographic, administrative	Methodology to rebuild per-disease and per-hospital	Respiratory illness patients from Quebec hospital from 1996 to 2006.	NB and Decision trees to predict readmissions	AUC
Braga et al.	Unknown	Labs and administrative	Methodology to rebuild per-disease and per-hospital	ICU patients of Centro Hospitalar do Porto, Portugal	NB to predict readmissions	Accuracy
Shams et al.	7200 instances	Demographics, socioeconomic, administrative	Methodology to rebuild per-disease and per-hospital	2011–12 Veterans Health Administration	RF and SVM to predict readmissions	Sensitivity, Specificity, F-score, AUC
Futoma et	3.3	Demographi	Methodolo	New	RF and	AUC

al.	million instances	cs and administrative	gy to rebuild per-hospital	Zealand National Minimum Dataset	SVM to predict readmissions	

Table 2.5 Summary of Modern PHRS

2.3.1 Localization

Scoring systems are often created once and available for use by any hospital. Hospitals may have differing patient demographics which make direct application of scoring systems difficult. LACE, for example, performs well on the dataset from which it was built. However, researchers have noted LACE may have poor performance when applied to hospitals with differing patient demographics.

Building separate models for each hospital may solve this issue [14]. Creating a scoring system such as LACE on a per-hospital basis may be time consuming and impractical. Machine learning algorithms use statistical techniques to build predictive models algorithmically. Thus, the importance of this type research is not the resulting model, but the methodology for which the model is created. Medical facilities wishing to implement a machine learning based PHRS need only supply the system with a set of training data. This allows localized models to be created with little difficultly. Many machine learning algorithms exist and have been successfully incorporated into PHRS to varying degrees. A summary of score-based and localized methods is outlined in Table 2.5.

A computationally fast algorithm often used for PHRS is Naïve Bayes (NB) [23]–[26]. NB is based upon Bayes' theorem and assumes conditional independence between features. While this may often not be the case with a given dataset, many times NB is still able to classify with sufficient performance [27]. If the NB conditional independence assumption holds, NB is able to converge using fewer instances than many other models

[28]. NB performs competitively against other algorithms for PHRS and has the advantage of quick model creation and classification time [24], [26].

Random Forests (RF) are a type of ensemble learning. RF is part of a family of algorithms known as decision trees. A major advantage of RF is that it is less prone to overfitting than many other decision tree algorithms [29]. RF accomplishes this by averaging multiple decision trees. An advantage of most decision tree algorithms are they can be interpreted visually as a graphical tree structure. However, RF loses this feature since many trees are used in the final model. RF works by applying sampling techniques to both instances and features. RF have been considered for inclusion in PHRS and is often the best performing algorithm [26], [30]–[33]. RF can be computationally intensive to build and slow to train. Once trained however RF is often quick to classify new data.

Support Vector Machines (SVM) are a type of classifier which creates a hyperplane that attempts to maximize the margin between classes. SVM is able to map inputs to higher dimensional feature spaces, thereby enabling it to do non-linear classification [34]. This gives SVM additional flexibility linear classifiers do not possess. SVM also have strong statistical theoretical foundations that many other classifiers do not possess and whose model is the global optimum. Although SVM often performs well in many other domains, it has not proven to be appropriate for PHRS [16], [31].

2.3.2 Disease Specific Methods

Readmission rates and feature distribution varies among diseases, as shown in Figure 2.1. This has motivated researchers to build disease-specific PHRS using similar methods to hospital-specific PHRS. These systems often attempt to select variables most appropriate to a given disease. Heart Failure (HF) has been the target of many PHRS [11],

[35]–[39]. Work by Wettersten et al. found that four variables commonly collected during hospitalization to have good predictive abilities [37]. These variables are B-type Natriuretic Peptide levels, number of medications prescribed at discharge, prescribed hypertension drugs, and blood pressure. Work by Thavendiranathan et al. introduced Echocardiogram readings as a variable [39].

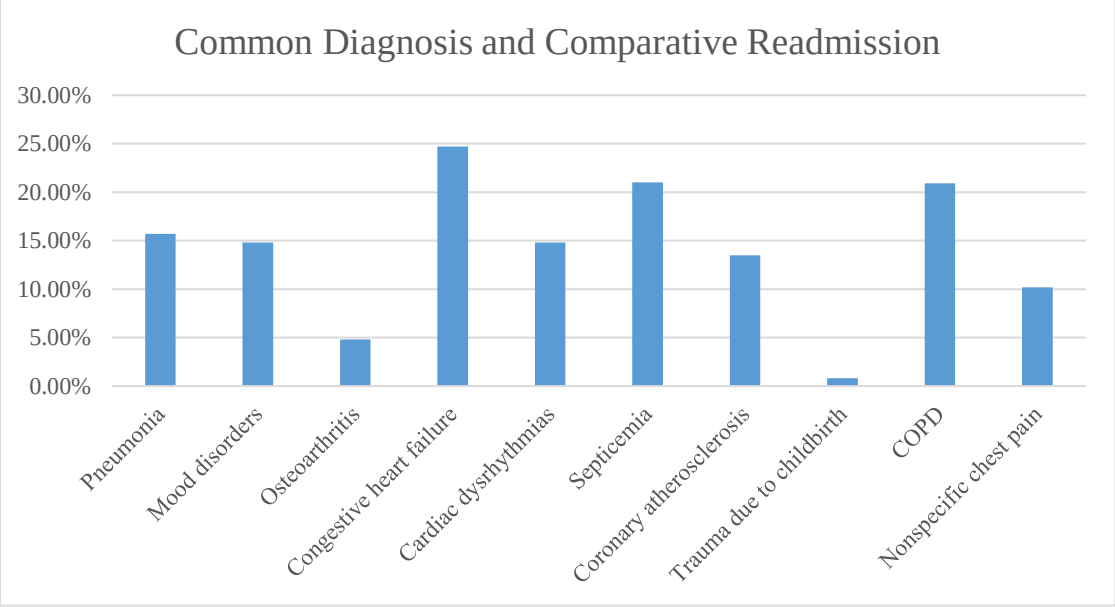

Figure 2.1 Common Index Diagnosis and Readmission Rates Vary Significantly [40]

Lung diseases such as COPD have also seen disease-specific PHRS research. Work by Agarwal et al. has incorporated smoking status as a variable in a PHRS [41]. Amalakuhan et al. have incorporated the frequency of COPD exacerbations as well as prescription of antibiotics. COPD patients often suffer from lung infections and use of antibiotics may indicate the patient experienced a severe exacerbation. Lee et al. have created a PHRS using a common COPD assessment test and spirometry data. The assessment test is a questionnaire containing questions such as “I cough all the time” and “My chest often feels tight.” Spirometry is a lung fitness test which measures the amount of air a patient inhales and exhales.

Patient surgeries are additionally a concern for hospital readmission. These patients are at an increased risk for infection and postoperative complications which often requires hospitalization [42], [43]. Fry et al. created a PHRS for predicting readmission of bowel operation patients [44]. Work by Kiran et al. also focuses on intestinal surgery readmissions [45]. Along with generalized variables used by many PHRS, urgency of surgery, complications during surgery, white blood cell count at discharge, and use of postoperative antibiotics were additionally introduced as variables.

2.3.3 Unstructured Data Sources

Recent works have begun to consider physicians notes as a primary data source [41]. The field of Natural Language Processing (NLP) attempts to extract usable information from unstructured text. Often this natural language is written English or other spoken language. Work by Duggal et al. uses NLP techniques to structure data and apply machine learning techniques to predict readmission. This system was able to achieve a maximum c-statistic of 0.69. The primary motivation of this research was the limited use of standardized data representations for diagnostic codes in Indian hospitals. Using clinical notes allows researchers to create a PHRS loosely coupled to the underlying patient record system. Domain experts and literature were consulted in selecting features to include. Figure 2.2 shows the final features to be included in this model. The primary patient population for this research is diabetes patients and 9,831 instances were available.

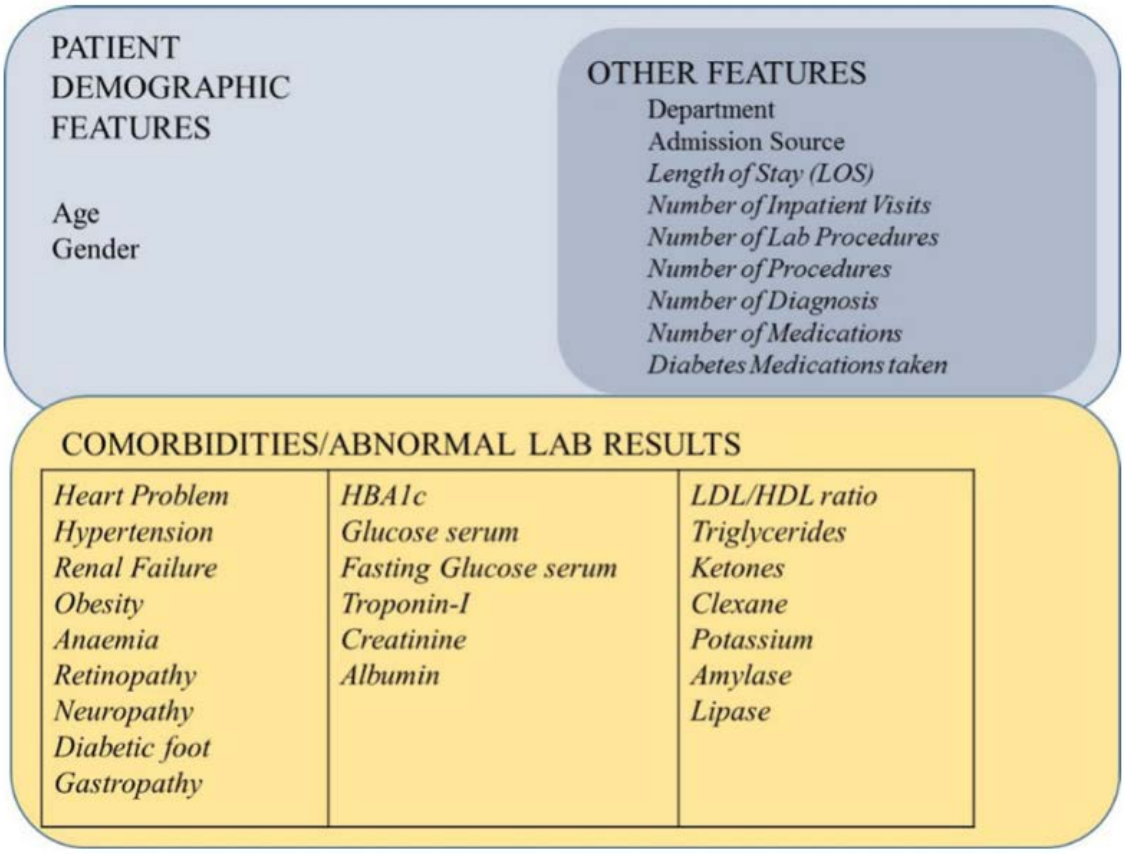

Figure 2.2 Features Used in NLP PHRS

Although this system represents significant progress over previous systems, the clinical note is used to primarily extract data traditionally already available in patient record systems. Other potential useful information may be discarded. Additionally, selecting useful features for each disease may be a time consuming and expensive task with little guarantee the selected features are optimal. Finally, as an early NLP based PHRS, researchers could have created and validated a model using only data derived from NLP without the introduction of structured data to compare the predictive power of purely NLP based data.

2.3.4 Cost as a Metric

Many modern systems use AUC as the primary metric [18]. However, cost may be

a more appropriate measure [46]. Predictive analytics regarding cost is considerably less studied than AUC. Research by Lawson et al. attempts to use Medicare claims data to analyze the cost of postoperative complications [43]. Data from the American College of Surgeons National Surgical Quality Improvement Program (ACS-NSQIP) were analyzed and a regression model created. The research focused on the hypothesis that surgeries with postoperative complications would have a higher 30-day readmission rate and increased cost. Additionally, researchers hypothesized that lowering complications could be associated with lowering cost.

Results found that reducing the ACS- NSQIP complication rate for each procedure by a relative 5% could result in the prevention of 2092 (95% CI = 1984–2201) readmissions per year and a savings to Medicare of $31.0 million (95% CI= 28.9–33.2). This cost savings is due to both the decreased number of readmissions occurring and the decreased cost of readmissions that occur despite the absence of a postoperative complication. A 20% relative reduction in the complication rates could result in the prevention of 8369 (95% CI = 7939–8800) readmissions and savings of $124.2 million (95% CI = 115.4–133.0). Finally, preventing all ACS-NSQIP complications for these procedures could result in the prevention of 41,846 (95% CI = 39,689–44,004) readmissions and a savings to Medicare of $620.3 million (577.4–663.2).

This work stops short of using statistical learning to predict cost for each patient. Research by Sushmita et al. attempts to go further than Lawson et al. and predict patient cost of readmission [16]. This work uses Mean Absolute Error (MAE) as the primary performance metric for numeric predictive models. This differs from classification tasks in that a cost associated with readmission, rather than readmission itself, is predicted. The

data used is from a large hospital chain in the Northwestern United States. Linear regression, M5 model tree, generalized boosted model, and decision trees were compared. M5 model tree was shown to have the lowest MAE. This work stops short of the previous analysis performed by Lawson et al. which analyzed the effect of lowering costs. Work by Carey et al. uses similar methodology to predict readmission costs in Veteran's Administration (VA) hospital [47].

2.3.5 Co-morbidities

Zeng et al. have created a system to assist in the detection of co-morbidities in clinical notes [48]. This system primarily uses HITEx to assist in the finding of co-morbidities. The existence of COPD and another disease in a clinical note is considered a comorbidity. This methodology is common in the determination of co-occurring diseases. However, this methodology may not be ideal as diseases which occur with high prevalence in a general population will statistically also co-occur with high frequency independent of COPD status. Ideally, penalizing diseases, medications, and symptoms which occur with high frequency in a general population would allow a more accurate picture. While such penalizations have been greatly researched in the Information Retrieval (IR) community [49], few have attempted to adapt these methods to clinical NLP [50], [51].

2.3.6 Big Data

Data mining of clinical data has been well studied since the emergence of the field. However, big data approaches have been far less studied. A review by Herland et al. documents several big data systems [52]. Current clinical big data systems are often built for the purpose of supervised machine learning tasks. In practice, many medical professionals use Clinical Decision Support Systems (CDSS) as these allow the

practitioner to make conclusions rather than relying on algorithmic classification. Many classification algorithms provide evidence which is difficult to interpret and medical professionals may be uncomfortable diagnosing patients without clearly interpretable results. Aggregated data, summary statistics, and similar patients are examples of common information presented to medical professionals using a CDSS. Big data approaches in CDSS have seen little research to date [53].

2.4 Summary of Existing PHRS

Current systems have several basic components in common. Score based systems are easy to implement but often perform poorly. This is due to differing data characteristics. Creating a model for each dataset appears to be most effective. Systems using machine learning generally perform better, but few systems have explored algorithms outside of the GLM family of algorithms. Systems using machine learning outline a methodology for building a system and each system implementation is created from training data.

Feature selection is important and using algorithms such as CS can potentially find an optimal subset of features. However, feature selection on a relatively small number of variables may be unnecessary and typically used for creating risk based scoring systems. Feature selection may be useful if a potentially large number of variables are available, however none of the reviewed systems were highly dimensional.

Data sources are additionally important. Creating systems from a general hospital population often results in lower performance. However, this may be misleading as some diseases have high readmission rates, leading to higher c-statistic. Performance of systems is often difficult to compare as each system is created and validated on different data. Clinical notes may offer easier implementation, but have been largely unexplored. Current

systems using clinical notes as a primary source have only used them in a limited manner, suggesting that additional useful information may be included in the model.

CHAPTER 3: BACKGROUND

3.1 Introduction

Several methods employed in subsequent chapters require foundational background information. Three primary datasets are used in this research, two of which contain clinical notes in the form of discharge summaries. NLP techniques for the structuring of unstructured text are outlined. Additionally, predictive modeling and subsequent evaluation is performed in many chapters. Algorithms and tools related to this are detailed in this background.

3.1.1 NLP

NLP has emerged as an important area of research in the field of data mining. Although structured systems such as relational databases have proven invaluable for fast and accurate data retrieval, vast amounts of information are stored in natural text such as the English language. Special care must be taken in the design of structured data storage and retrieval systems and data must often be recorded in an unnatural manner. Humans naturally write words, sentences, and paragraphs to convey information. Natural language may be a more familiar method to express information than structured systems, but is considerably more difficult to data mine. Sophisticated algorithms have been developed to structure such data, but are frequently less accurate than structured systems.

3.1.2 History

The history of NLP spans many centuries and has been a research topic for numerous prominent linguists. After the creation of practical computer systems post World

War II, NLP was dominated by structured formal grammars whose rules can be programmed into a computer [54]. These grammars were typically formed by hand written rules developed by linguists. Many theories regarding these grammars formed and dominated the field for more than fifty years.

Beginning in the late 1980's the focus of NLP changed from hand written rules to machine learning algorithms. The theories of many linguists of the time encouraged a universal understanding of language. However, machine learning techniques employed the idea of corpus linguistics which studies real world pieces of text to develop abstract rules. These rules may only apply to text in the same domain as the training corpus. Systems involving hand coded rules are often fragile and require expert linguists to formulate. Statistical approaches using machine learning also have the advantage of assigning probabilities to possible outcomes. These probabilistic approaches dominate the field today.

3.1.3 Major Tasks

The creation of practical NLP systems often involves many tasks. Several high-level NLP tasks cannot be performed without prerequisite tasks being completed and will often be structured into an NLP pipeline. Figure 3.1 shows a typical NLP pipeline for basic document structure as described by Bird et al. [55]. One of the lowest level tasks in this pipeline is tokenization. Tokenization is the process of splitting a document into basic linguistic units called tokens. For the English language, this can be relatively straight forward. Trivial cases may be solved by simply splitting words based on non-alphanumeric characters [56]. However, tokenization quickly becomes more complex when outlier cases are encountered.

As an example, the word *co-education* would be tokenized into two tokens when splitting on non-alphanumeric characters. This may not be the desired result. Introducing an exception for the hyphen character might not suffice. A commonly employed word grouping device may warrant splitting, as is the case for *copy-on-write.* Contractions such as *can't* and *aren't* are equally ambiguous. It becomes quickly apparent this simple model breaks down when presented with non-trivial cases.

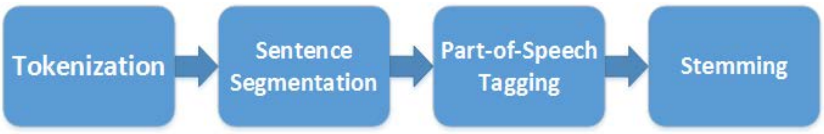

Figure 3.1 Typical NLP Pipeline for Low Level Processing

Sentence segmentation is similar to tokenization and offers many of the same challenges [57]. In English, the period character will often denote the end of a sentence. However, there are many exceptions. *Dr. Smith returned to his office later that morning*. is an example where a period would not denote the end of the sentence. Some systems may attempt to create hand crafted rules for sentence segmentation and performance can be acceptable. Using the following rules, researchers at data analytics firm Attivio were able to detect 95% of sentence segments. (1) If it is a period, it ends a sentence. (2) If the preceding token is in a list of known abbreviations, it does not end the sentence. (3) If the next token is capitalized, it ends the sentence [58].

Part-of-speech (POS) tagging is the process of marking words of a document as a given part of speech [59]. The parts of speech used may vary greatly between systems. A simple system may only be concerned with nouns, verbs, and adjectives, whereas a complex system such as the Penn Treebank uses 36 different parts-of-speech. An example of a tagged sentence is shown in Figure 3.2. Various approaches for part of speech tagging

exists, and as with other tasks in the pipeline, fall into two categories: rule based and machine learning.

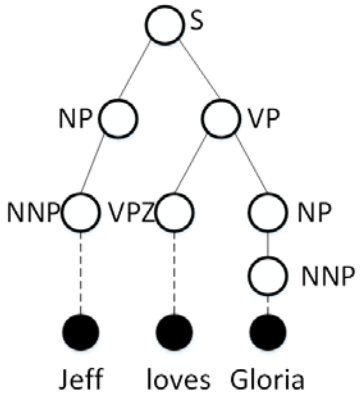

Figure 3.2 POS Tagged Sentence Using Treebank Representation

Many words can be represented via several inflections and can have many derivations. Stemming is the process of recognizing various forms of a word and reducing it to its base form [55]. For many applications, the presence of any form of the word is more important than which form it has taken. The forms *fishing, fish,* and *fisher* would be reduced to the base form *fish.* Other examples are not as simple. The forms *argue, argument, arguing,* and *argues* reduce to the base form *argu* which may not seem intuitive.

One of the most popular stemming algorithms is the Porter stemmer created by Martin Porter in 1980 [60]. The algorithm is a set of rules that can be summarized as six basic steps.

1. Get rid of plurals, and –ed and –ing suffixes.
2. Replace *y* with *i* if there is another vowel in the stem.
3. Map double suffixes to single suffix.
4. Remove –full and –ness suffix types.
5. Remove –ant and –ence suffix types.

6. Remove a trailing –e

The original Porter stemmer was written in BCPL, a language not in widespread use today. This caused programmers to create implementations in several other languages. These implementations were of varying quality and contained errors. To combat this, Dan Porter later wrote a version in C to act as the reference implementation and the Porter stemmer has correct implementations in at least 25 different programming languages today [61].

Hand crafted rule systems have many limitations. They are often complex, fragile, and must be constructed by linguistic experts. Machine learning techniques offer a solution to these problems. Given an annotated set of documents, machine learning techniques can construct models which are able to make future predictions of document structure. Although annotating document structure typically requires human oversight, many times the annotation tasks can be completed by anyone fluent in the language and need not be a linguistic expert.

A very popular machine learning algorithm for learning structure is Maximum Entropy (ME). ME is the standard algorithm for tokenization and sentence segmentation in many NLP toolkits and has applications in other NLP tasks such as POS tagging and NER [62]. As with many machine learning algorithms, a prerequisite for ME is that features must be extracted from the document.

For example, when creating a model which performs sentence segmentation, features which relate to sentence structure are chosen. A feature to be extracted may be whether or not the word contains a period. Words containing a period would often be a good indicator of the end of a sentence. Mathematically this would be represented

$$f_i(c,d) = [c = SENTENCE \ \wedge hasPeriod(d_w)]$$

Where c is the classification, d is the data available, and d_w is the current word under analysis [63]. Many features can be chosen in a model and f_i represents the i^{th} feature. This can be generalized to

$$f_i(c,d) = [c = c_j \ \wedge \ \Phi(d)]$$

Where c_j is a particular instance of a class and $\Phi(d)$ can be any valid predicate. Although the function can return any value in $\mathbb{R}$, in practice the function usually returns a binary value. The process of selecting these features is known as feature engineering and the quality of features selected directly impacts the performance of the algorithm.

Once features have been extracted, the ME algorithm can be utilized. Each possible class is voted and the most likely class is chosen. In the sentence segmentation example, the two possible classes would be SENTENCE and NOT_SENTENCE. The following equation describes voting.

$$vote(c) = \ \Sigma \lambda_i f_i(c,d)$$

Where λ_i is a weight chosen for that instance of a feature. Weights are chosen to maximize the conditional likelihood of the data according to the model and can be found using optimization algorithms.

3.2 NLP Libraries and Frameworks

NLP is an ongoing research topic that has seen many systems developed. Early research systems implemented NLP tasks without the assistance of software libraries. As the field matured, libraries and toolkits became available. These software components are aimed at being reusable so that well studied tasks such as tokenization are not implemented from scratch each time a system is developed. These software components fall primarily

into two categories: libraries and frameworks.

3.2.1 NLP Libraries

Software libraries are generally defined as a collection of routines, functions, or classes which are designed to abstract a complex problem. They are created with reusability in mind and designed to enable programmers to write software without duplication of efforts. NLP has many libraries available, aimed at different languages and purposes.

3.2.1.1 OpenNLP

OpenNLP was first created in 2000 as a set of Java interfaces meant to create a standard API for common NLP tasks. The original implementation of these interfaces was created by researchers at the University of Edinburgh in a system known as Grok [64]. In 2010 the project was incorporated into the Apache incubator where the interfaces and implementation were merged into a single toolkit. In 2012 OpenNLP graduated to an Apache top-level project.

The goal of OpenNLP is to provide a set of libraries for well-studied NLP tasks such as tokenization, sentence segmentation, part-of-speech tagging, named entity recognition, and stemming [65]. The toolkit uses a machine learning approach for most tasks rather than a set of hand-crafted grammar rules. OpenNLP offers command line tools and API's for creating models and testing their performance. To train these models however requires documents be annotated manually as most of its learning algorithms are supervised. For users without the resources to annotate training data, OpenNLP provides models trained on several popular corpora including the Brown corpus and Reuters corpus.

3.2.1.2 Stanford CoreNLP

Stanford CoreNLP is a library aimed at integrating Stanford's many NLP research

projects. CoreNLP has a great deal of overlap in functionality when compared to OpenNLP and aims to provide low level tasks such as tokenization to high level tasks such as coreference resolution [66]. CoreNLP is written in Java and was released in 2010 by the Stanford Natural Language Processing Group.

3.2.2 NLP Frameworks

Often times various NLP systems have very similar designs. Higher level tasks such as NER depend upon lower level tasks such as tokenization. To avoid repeatedly designing NLP systems from the ground-up, several frameworks exist. Frameworks are very similar to libraries in that they both intend to produce reusable systems. Frameworks may even use libraries and make libraries available. The key difference between libraries and frameworks are that frameworks rely on Inversion of Control (IoC) [67]. In a typical computer program, the entry point of the program is code that the user has written and the flow of code executed is determined by the user's code. Programs that rely on frameworks generally provide sets of routines available to the framework and the framework determines when and how to call those routines.

Frameworks become useful for NLP processing because a frequent design pattern used in NLP is that of the pipeline pattern [68]. NLP tasks are often arranged from low level tasks to high level tasks with each task possibly depending on the previous. For example, tokenization is generally an initial task in the pipeline, then sentence segmentation, then POS tagging, then stemming, with each task depending upon information from the previous. Using a framework, users can assemble a pipeline of tasks specific to the goal of the system. Several implementations of NLP frameworks using the pipeline design pattern exist.

3.2.2.1 Apache UIMA

Apache Unstructured Information Architecture (UIMA) is a framework that started at IBM research in 2004 to address the growing need to structure large systems that processed unstructured data [69]. At the time, IBM had over 200 researchers and developers working on Unstructured Information Management (UIM) projects. Research groups were duplicating work and at the time there existed little means to quickly integrate others' code. UIMA was created with the goal to write small routines of code that could be reused. These routines are known as *annotators*. Each annotator is run serially in a pipeline and given metadata from the previous annotator before execution. Each annotator must be placed in the pipeline where it can be executed with all annotator dependencies met.

The metadata associated with each document is known as the Common Analysis System (CAS). The CAS provides a standard set of types and ability to declare custom types to be used. Each document has exactly one CAS. UIMA is designed to be language agnostic and as of this writing annotators can be written in Java, C++, Perl, Python, and Tcl [70]. In practice, many systems are written purely in Java and there exists a wrapper around the CAS with several convince methods known as JCas. The pipeline of annotators is known as the Analysis Engine (AE). An AE can be composed of other AE to simplify pipeline creation and remove redundancy in declaring similar pipelines.

In 2005 IBM declared UIMA an open source project and in 2010 UIMA become a top level Apache project [71]. Several additional projects have been created to simplify UIMA pipeline and component development and to enhance scalability. The uimaFIT project was originally created in 2008 to assist in the simplification of testing UIMA components [72]. The project quickly expanded to simplify pipeline creation and type

system declarations. Apache UIMA uses a series of XML files to define AE pipelines and CAS types. These XML files in turn generate code in the programming language of choice. UimaFIT allows types and AE to be declared in code rather than XML configuration files. This can often greatly reduce the number of files in a UIMA project and reduce complexity. It has the ability to generate XML files from code based declarations if the need for XML files should arise. As of this writing, uimaFIT is only compatible with Java, but since it is possible to generate UIMA XML files, any UIMA compatible language can still be used.

3.2.2.2 GATE

General Architecture for Text Engineering (GATE) is a framework aimed at both offering pipelines and libraries for NLP tasks. This allows a single software system to provide a comprehensive solution to many NLP tasks. GATE was created in 1995 at the University of Sheffield and is written in Java. A graphical user interface allows users to build a pipeline visually without writing any code and can be used by non-programmers. Other frameworks and libraries listed are aimed mostly at programmers building systems rather than end users.

GATE includes a basic pipeline called ANNIE which contains tokenization, sentence segmentation, POS tagger, and NER. A plugin system called Collection of Reusable Objects (CREOLE) allows this basic functionality to be expanded and a central repository allows developers to make their custom plugins available [73]. In addition to pipelines and NLP tools, GATE offers tools for information retrieval such as full text searching and concept searching using ontologies.

3.3 Medical NLP

The medical domain has been one of the earliest applications of NLP. Medical

professionals often write clinical notes which summarize a patient's condition, medications, labs, treatment course, family history, and anything else deemed important. Patient records generally include structured medical information in addition to unstructured text. However, this information is usually meant for billing purposes and to comply with state and federal reporting laws. It is not meant to convey a complete picture of the patient. While there is not agreement as to exactly how much data is stored in unstructured format, reports agree much of the information is kept in unstructured documents [74]. The reason so much medical data is not structured is additional office staff known as medical coders must translate the medical expert's notes to structured form. This translation is costly and often times only the bare minimum needed for processing is performed. Thus, NLP offers a method to possibly extract a great amount of information that is not captured in structured notes.

3.3.1 UMLS

The Unified Medical Language System (UMLS) is a collection of controlled vocabularies in the clinical domain [75]–[77]. The collection began in 1986 by the National Library of Medicine (NLM) and provides various mapping mechanisms. Mappings between vocabularies are available and provide synonyms of many medical terms normalized to a single form. Additionally, semantic structure is available offering many different viewpoints of medical terms.

UMLS Metathesaurus is most commonly used in this research. This system comprises over 1 million biomedical concepts and 5 million concept names. Over 100 controlled vocabularies (often referred to as dictionaries by clinical NLP systems) are available. This database is updated quarterly and is available free for non-commercial use.

Additional tools and libraries are available to support UMLS Metathesaurus. MetamorphoSys is a graphical application that allows users to choose the vocabularies for which to download and include in a SQL database. LVG is a Java library that generates lexical variants of a given term and offers additional clinical NLP support for common tasks related to stemming. These tools offer support for clinical NLP tasks, however are often based on substring matching. This is in opposition to many systems which offer probabilistic approaches. The NLM actively maintains these tools and databases which may become outdated as new medications and disease terminology are introduced into the clinical domain. In practice, most modern clinical NLP systems use UMLS due to the active maintenance by the NLM.

UMLS offers semantic mapping through UMLS Semantic Network (USN). USN allows relationships between medical terms to be extracted for data mining purposes. A common problem in NLP is stemming, which is often solved by the porter stemmer. However, medical terms may have complicated relationships for which the porter stemmer is appropriate. For example, a single disease may have many abbreviations and historical terms. USN offers mappings among medical terms to a single concept. Figure 3.3 shows several variations for hypertensive disease normalized to a common UMLS Concept ID (CID). Normalization to a common ID is necessary as variations in the same medical term will in terms with lower than appropriate representation.

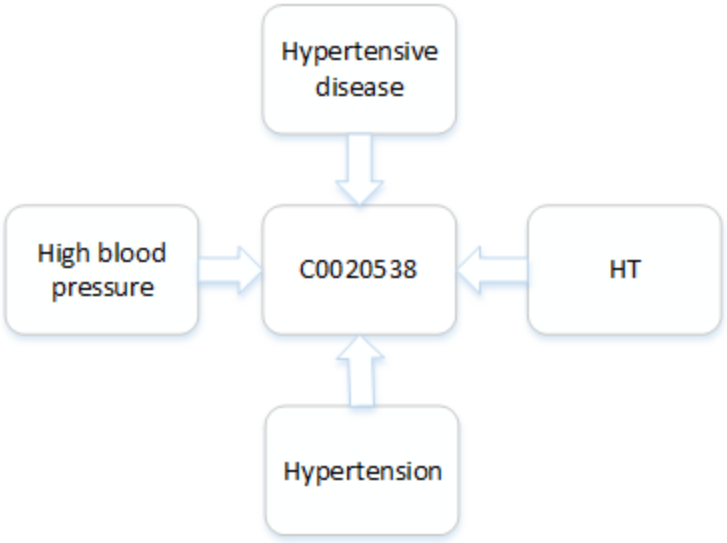

Figure 3.3 Normalization of Hypertension and its Variants to a Single UMLS CID

The UMLS dictionaries used in this research are Snomed-CT, NCI Thesaurus, MeSH, and ICD-9. Table 3.1 outlines a brief description of each dictionary. Many terms are identified in multiple dictionaries and there exists a large amount of redundancy in coverage. UMLS CIDs are valid across dictionaries and it is possible to normalize a term discovered in multiple dictionaries to single CID. The use of multiple dictionaries assists in expanding coverage of common abbreviations and variants in spelling.

Dictionary	**Description**
Snomed-CT [78]	Snomed-CT is a set of clinical terms maintained by the International Health Terminology Standards Development Organization (IHTSDO). The development of Snomed-CT dates back to 1965 and is known for its comprehensive coverage of clinical terms. SNOMED-CT consists of concepts, descriptions, and relationships and can be used for semantic processing.
NCI Thesaurus [79]	The National Cancer Institute (NCI) Thesaurus was created to assist in research systems made available by NCI and scope is to cover clinical terminology covering cancers, findings, drugs, therapies, anatomy, genes, and many other cancer research related terms. NCI Thesaurus offers a partial model as to how these subjects relate to each other and aims to provide a common system for cancer researchers to communicate.
MeSH [80]	Medical Subject Headings (MeSH) is an NLM controlled vocabulary used for indexing articles on NIH's Pubmed.

	Additionally, relationships between terms are provided which can act as a thesaurus.
ICD-9 [81], [82]	International Classification of Diseases (ICD) is a coding system designed for classification of diseases and disorders. ICD is maintained by the World Health Organization (WHO) and ICD-9 is the ninth revision of the system. In the United States, ICD-9 has seen popular usage in medical billing. The system has been adopted by many organizations, including the Centers for Disease Control for reporting mortality and morbidity statistics [83].

Table 3.1 Summary of Dictionaries Used by cTAKES

3.3.2 LSP-MLP

The oldest traceable NLP system directed at information extraction of clinical notes is the Linguistic String Project – Medical Language Processor (LSP-MLP). LSP began in 1965 for the purposes of developing an English language parser that could process scientific literature [84]. The goal for LSP was to structure text in a way that could answer questions based on structured queries. After initial successes in the scientific literature domain, the National Institutes of Health (NIH) funded an expansion of LSP to be applied to clinical narratives. This work resulted in the MLP system and was aimed at supplementing LSP with domain specific knowledge. These supplements are mainly in the form of medical vocabularies provided by the National Library of Medicine (NLM). However, research has stalled since the late 1990's and there have been no new publications related to LSP-MLP since.

The LSP-MLP system proved to be quite useful for its time period. However, several NLP developments have superseded LSP-MLP usefulness. Modern techniques typically use a statistical approach for the identification of parts of speech, sentence boundaries, and other structure. LSP attempts to structure language using grammars. Research has shown machine learning techniques using Maximum Entropy (ME) to have superior performance to a grammar based approach for many situations. MLP does not use

a controlled medical vocabulary and is unable to map expressions to codes or normalized forms. The system is also fragile because it depends highly on the syntactic structure of the text rather than semantic meaning. Small changes in sentence syntax yield different extraction results.

3.3.3 MedLEE

The Medical Language Extraction and Encoding System (MedLEE) originated as a system to structure radiology reports. MedLEE eventually evolved as a general purpose system for clinical notes not limited to radiology reports [85]. Processing of clinical narratives is constructed in a pipeline. MedLEE was created when pipeline frameworks such as UIMA and GATE were not available and created a pipeline from the ground up.

In 2000 MedLEE improved to include medical vocabularies and was one of the first medical NLP systems to do so [85]. The National Library of Medicine's Unified Medical Language System (UMLS) provides medical terms from many data sources in a standardized manner and MedLEE included dictionaries from SNOMED and ICD-9. In 2012 MedLEE began a commercial partnership with Health Fidelity for inclusion in their risk management software. MedLEE structures clinical notes to be used as a data source for higher layers in Health Fidelity's DISCERN product [86].

3.3.4 HITEx

Health Information Text Extraction (HITEx) is an information extraction system aimed at general purpose processing of medical texts. The system departs from previous works as it uses a component based architecture based on the GATE framework [48]. The use of GATE allowed researchers to focus solely on domain specific concerns rather than low level tasks such as tokenization and sentence segmentation. This project has stalled

and not published new research since 2006.

The highest layer of HITEx is the UMLS concept mapper which maps medical terms to UMLS concepts. This subsystem uses both exact string matching and fuzzy matching through truncation and normalization. The system has been used successfully as a hybrid system combining ICD-9 structured data and unstructured clinical notes. In addition to basic NLP tasks, HITEx contains modules capable of discerning the patient's primary diagnosis and smoking status.

3.3.5 cTAKES

The Clinical Text Analysis and Extraction System (cTAKES) was created by researchers at the Mayo clinic beginning in 2006 and is still actively maintained. cTAKES uses a component based architecture and is based on IBM's UIMA [87]. cTAKES uses Commercial Off The Shelf (COTS) software components for many parts of the system. Apache OpenNLP and Stanford CoreNLP provide functionality for low level NLP tasks such as tokenization, sentence detection, chunking, part of speech detection, and other common NLP tasks. cTAKES uses UIMA Annotators to extract basic NLP information from the document and add the information to the CAS. A summary of cTAKES components is provided in Figure 3.4.

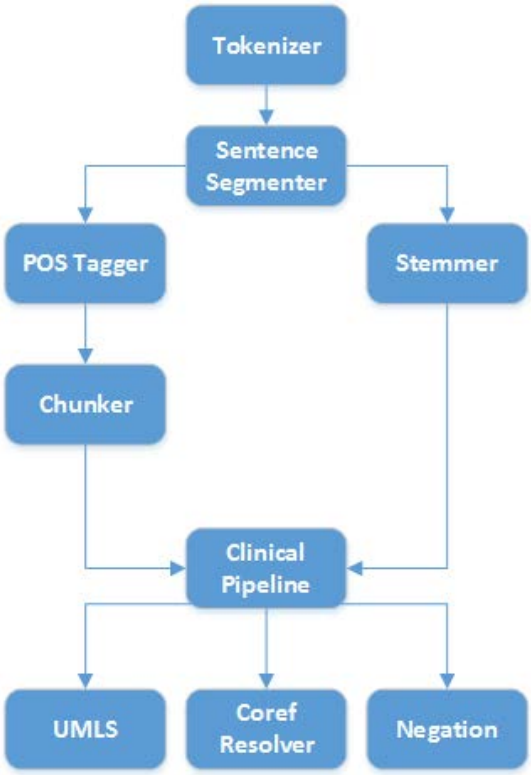

Figure 3.4 cTAKES Components

cTAKES uses ICD-9, SNOMED-CT, NCI Thesaurus, MeSH, and RxNorm dictionaries. Medical term matching does not use probabilistic approaches and instead uses substring matching. In addition to dictionary based matching, cTAKES is able to discern positive and negative conditions, temporal events, and distinguish between conditions affecting patient vs family histories. The project is still actively maintained and regularly participates in i2b2 competitions.

3.4 Feature Engineering

In order to build a predictive model, features must be extracted from unprocessed data. A feature is an individual measurable property of the phenomenon being observed. The field of machine learning which focuses on creating these features is known as feature engineering. This process can either be manual or algorithmic. Incorporation of domain

knowledge can increase the quality of these features and in turn increase the quality of the predictive model. Two distinct phases of feature engineering are utilized feature extraction and feature selection. Given a piece of natural language, several methods exist for extracting features.

3.4.1 Bag of Words

The bag-of-words representation method treats each word in the corpus as a feature. Each document is an instance and each word is either present or not-present in the instance. This method is simple and can be used with most any natural language document. No additional domain knowledge is required to prepare the data. However, several downsides exist. This method will often produce many features, typically in the range of 1,000 to 100,000 features. This greatly increases model creation time and can be expensive to represent in memory. Techniques such as sparse feature representation need to be employed to reduce memory requirements, increasing implementation complexity. Though simplistic, bag-of-words often produces good results with minimal feature engineering and can be combined with other techniques such as tf-idf to give unequal weighting based on document frequency.

3.4.2 cTAKES Annotation

Although bag-of-words can often be an acceptable method for feature extraction, many times this method is too simplistic. Bag-of-words makes no effort to use domain knowledge of a given piece of text. Analyzing the text and extracting higher level features is often desirable. For example, a single disease may have several common abbreviations and spelling variations. Using domain knowledge of these variations in spelling allows the feature to be reduced to a feature representing the presence of the disease rather than

presence of words.

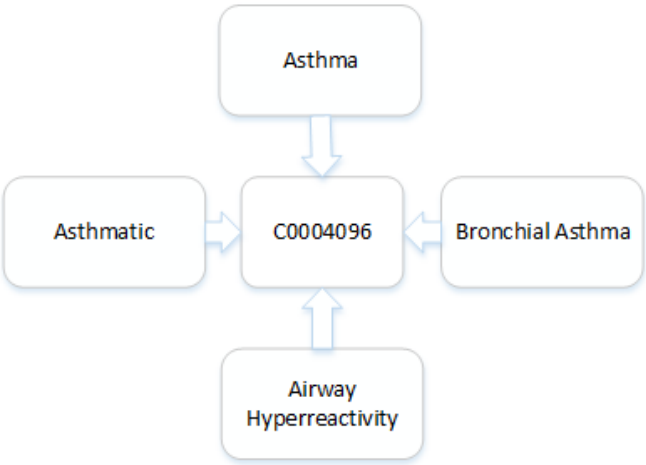

Figure 3.5 Various Forms of the CID C0004096 Representing Asthma

This higher level feature extraction is done using Apache cTAKES. Diseases and disorders are extracted and normalized using UMLS to a single CID. Figure 3.5 shows an example of a disease with several representations and its normalized form. Medications are additionally extracted and normalized to their common name.

3.5 Feature Selection

Inclusion of all discovered features in model creation often has drawbacks and may not be desirable. Bag-of-words representation is known to be highly dimensional and suffers from a phenomenon known as the "curse of dimensionality." Certain mathematical techniques used in low dimensional space become less effective in high dimensions. For example, distance functions can be used in the building of classification algorithms. Functions such as Euclidean distance are effective in low dimensions, but in high dimensions there is little difference in distance between points rendering the function of little use. Workarounds can often be found, such as using cosine similarity instead of Euclidean distance, but this may require modification of the classification algorithm which may not be feasible in many cases. Additionally, models which contain many features

typically run much slower than their low dimensional counterparts. Reducing the number of features can potentially result in a faster model with better classification characteristics.

A method used to reduce the number of features is known as feature selection. Ideally, removing features which offer little or no information to the classification algorithm is desired. Feature selection can be broadly categorized into three groups: (1) Filter (2) Wrapper and (3) Embedded. Filter methods use statistical tests to rank features by relevance. They are typically quick to compute compared to other methods but may not find an optimal set of features. Wrapper methods test all possible combinations of features with a fixed classification algorithm and use a performance metric such as accuracy to find the highest score. It may be possible to find the most useful features using wrapper methods, but this method is computationally expensive and will often lead to overfitting. Embedded methods have feature selection built into the classification algorithm. The C4.5 decision tree algorithm is an example of embedded feature selection as it uses Information Gain Ratio to select which features to use in building tree nodes. Filter and wrapper feature selection methods are evaluated in this research and four methods are analyzed to determine which is most useful for final inclusion in the framework.

3.5.1 Wrapper with Forward Selection

Wrapper based feature selection chooses a subset of features then builds a classifier from the reduced feature set. Although this method can work very well, it has several downsides. For each set of features chosen, a classifier must be built and tested. Some classifiers such as NB can evaluate a test set very quickly, but many others may require a non-trivial amount of time to build. To test all possible sets of features requires $2^n - 1$ iterations. Since the purpose of feature selection is often to reduce possibly thousands of

features, testing all possible combinations quickly becomes impractical.

An alternative method is to find a locally optimal set of features that works well. Forward selection is one variant of this selection method. Given a set of features, the algorithm builds and evaluates all possible models consisting of a single feature. The feature that performs best is chosen. The chosen feature is kept and evaluated with all remaining features that forms a pair of features. The best pair is chosen. This algorithm uses a greedy approach and iteratively runs until the features are exhausted. Termination may occur if the model does not improve as features are added or a threshold is reached.

3.5.2 Correlation Feature Selection

Correlation Feature Selection (CFS) is based on finding features which correlate highly to the class but does not correlate highly with other features. Pearson's correlation coefficient is used to test the correlation among features and the class variable. Irrelevant features are ignored as they have low correlation with the class. The inclusion of features depends whether or not an already selected feature contains similar information, thereby removing redundant features. The subset is given a score rather than tested by building and evaluating a classifier. This means there is generally a large speed improvement over wrapper feature selection methods.

3.5.3 Gain Ratio

Information Gain (IG) is a method often used in building decision trees (such as ID3) which measures the change in information entropy from a prior state that takes into account some information. Since decision trees often have built-in (embedded) feature selection, IG can be used in isolation as a filter method to determine which features are most useful. Gain Ratio (GR) is a modification of IG which penalizes bias towards multi-

valued attributes.

3.5.4 Chi-Squared

The Chi-Squared (CS) test is a statistical measure of the independence of two events. CS tests are often used in experimental research to perform hypothesis testing on two groups of data. When applied to feature selection, the two events under observation are the occurrence of the feature and occurrence of the class. The null hypothesis is the feature and class are independent.

3.6 Classification

3.6.1 Naïve Bayes

Naïve Bayes (NB) is a simple probabilistic classifier that is based upon Bayes' theorem. The classifier assumes independence between features. Though many times this independence assumption is not true, in practice NB still works well. NB uses little memory and can classify new instances quickly. Early methods for e-mail spam detection used NB due to this speed. NB is known to work well in text classification contexts and was chosen for this quality.

3.6.2 Random Forest

Random Forest (RF) is an ensemble algorithm which creates multiple decision trees by randomly choosing a set of features to use for each tree. RF can be conceptually thought of as bagging features. Bagging is an ensemble method which creates multiple classifiers by sampling dataset instances. RF uses a similar method, but for features instead of instances. An advantage to RF is that it is less prone to overfitting than many other decision tree algorithms.

3.6.3 K-Nearest Neighbors

K-Nearest Neighbors (kNN) is a non-parametric algorithm which uses a distance function to find the instances which are most similar to the current instance. In this research, Euclidean distance is used as the distance function. k is a variable which can be any number less than or equal to the number of instances in the training set. In the simple case where k=1, the classification of the nearest instance is taken as the classification of the instance under consideration. When k > 1, several methods exist to aggregate instances to output a single classification. The method used in this research is majority voting and k=3 so there are no tied votes.

3.6.4 Support Vector Machine

Support Vector Machines (SVM) create a hyperplane which attempts to maximize the margin between classes. This is achieved by selecting a small number of boundary instances and building a linear function which maximizes separation. Unlike some other linear classifiers, SVM is able to classify nonlinear class boundaries. SVM has the property of stability and does not change much when a small number of instances are added to the dataset and overfitting is unlikely to occur. Although SVM have many positive theoretical properties, training SVM can often be slow, especially in the case of highly dimensional datasets.

3.6.5 Ensemble Learning

Ensemble learning approaches leverage many classifiers in order to obtain a classification result. A mechanism which aggregates individual classification results is required. Majority voting is an often used mechanism which solves this problem. Two types of majority voting exist. Hard majority voting sums the binary classification results

of the base classifiers. The class label with the largest number of votes is assigned to that instance. Soft majority voting uses the summation of posterior probabilities to reach a classification result.

The two methods of ensemble learning used in this research are bagging and boosting. Bagging samples instances (with replacement) in order to create a representative dataset. A classifier is then trained based on the sampled dataset. This process is performed iteratively created many so-called *bags*. Bagging can help reduce variance and avoid overfitting. Boosting attempts to create learners which are experts at classifying subsets of data. The algorithm iteratively creates a set of classifiers. During each iteration, instances which were previously misclassified are given greater weight when building the next classifier.

3.7 Evaluation

Performance of classification systems must be evaluated to determine the systems usefulness. This evaluation is often in the form of common statistical measures. A classification system is generally trained using a subset of available data. A hold-out dataset containing unseen instances of data is used to evaluate its performance. Ground truth labels indicating the actual status of readmission are available, but withheld until final evaluation. The evaluator attempts to predict each instance of the hold-out dataset using the trained model. The prediction is then compared to the ground-truth classification label. There exist four possible outcomes for binary classification: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN).

An intuitive evaluation metric is accuracy, defined as follows.

$$Accuracy = \frac{\sum TP + \sum TN}{\sum \mathrm{TP} + \sum TN + \sum FP + \sum FN)}$$

However, this metric may be misleading and is only used by a small number of PHRS. Hospital readmission labels are often imbalanced because most patients are not readmitted within 30 days. A simple but useless classifier may choose to classify all patients as non-readmission. Applied to a dataset with a 10% readmission rate, this classifier would have 90% accuracy, but be of little use.

The most common performance evaluation metric for modern PHRS utilizes the Receiver Operating Characteristic (ROC). ROC consists of the True Positive Rate (TPR) and False Positive Rate (FPR). TPR and FPR are defined as follows.

$$TPR = \frac{\sum TP}{\sum TP + \sum FN}$$

$$FPR = \frac{\sum FP}{\sum FP + \sum TN}$$

For binary classification systems, a probability of class membership can often be produced. Some algorithms, such as NB, inherently use this probability for classification. Many other algorithms can coerce a posterior probability despite not requiring it for classification. By varying the threshold of classification, a ROC curve can be plotted. The Area Under the ROC curve is often referred to as AUC (also known as c-statistic) and is a most common metric for evaluating PHRS. Producing misleading results using AUC is considerably more difficult than it is when using accuracy. A system that produces very good TPR may produce an extremely poor FPR, lowering AUC.

CHAPTER 4: METHODOLOGY

4.1 Readmission Prediction using Natural Language Processing

Current systems often choose structured data as a primary data source and improvement of models is generally in the form of improved feature selection and engineering. However, using clinical notes as the primary data source has shown to potentially offer similar or better performance than structured data sources. PHRS has generally seen low adoption rates in hospitals and using clinical notes in the form of discharge summaries may potentially increase adoption since these are often easier to extract. Extraction of structured data may require the implementation of an HL7 engine or direct access to a SQL database. These methods may require a large IT integration project which may be both time consuming and expensive.

Clinical notes are often significantly easier to export from an EHR. Many times these notes are stored on a filesystem as a plaintext or PDF document and contain sufficient header data to identify a patient identification number, date of admission, and date of discharge. In consultation with domain experts a hospital administrator noted the capability of high speed printers and optical character recognition to require no involvement of IT staff and the potentially cheapest option. Although this is an extreme case, it underscores the preference for hospital administrators to use a method loosely coupled to the underlying EHR.

Early efforts attempting to use discharge summaries as a primary data source have had several shortcomings. Current systems require some structured data to be available. A

system which is able to use only the discharge summary without structured data support would be preferable as it would remain loosely coupled to the underlying EHR. Current systems are generally disease specific. A general purpose approach would reduce time consuming efforts using domain experts to customize each system. Finally, current systems are either annotation based or word based. Using all available data, both annotation based and word based, may increase model performance.

4.1.1 Framework

The framework for predicting hospital readmissions using clinical notes is composed of several components. Each component is evaluated in order to present the optimal algorithm selection. Initial preprocessing of clinical notes may be necessary to produce an ASCII text document. Feature extraction is comprised of two methods: (1) cTAKES annotation and (2) bag of words representation. Useful information may be contained within the clinical note that is not annotated within cTAKES. Therefore, both representations are merged and feature selection is performed.

Both wrapper and filter feature selection methods are chosen for this framework. Wrapper CFS, GR, and CS are chosen for feature selection. Feature selection serves to improve classifier execution time. The number of selected features is increased iteratively and the effect upon model performance are evaluated. SVM, RF, NB, and kNN classification algorithms are evaluated. Evaluation is performed using AUC and time. A pipeline of component evaluation is shown in Figure 4.1.

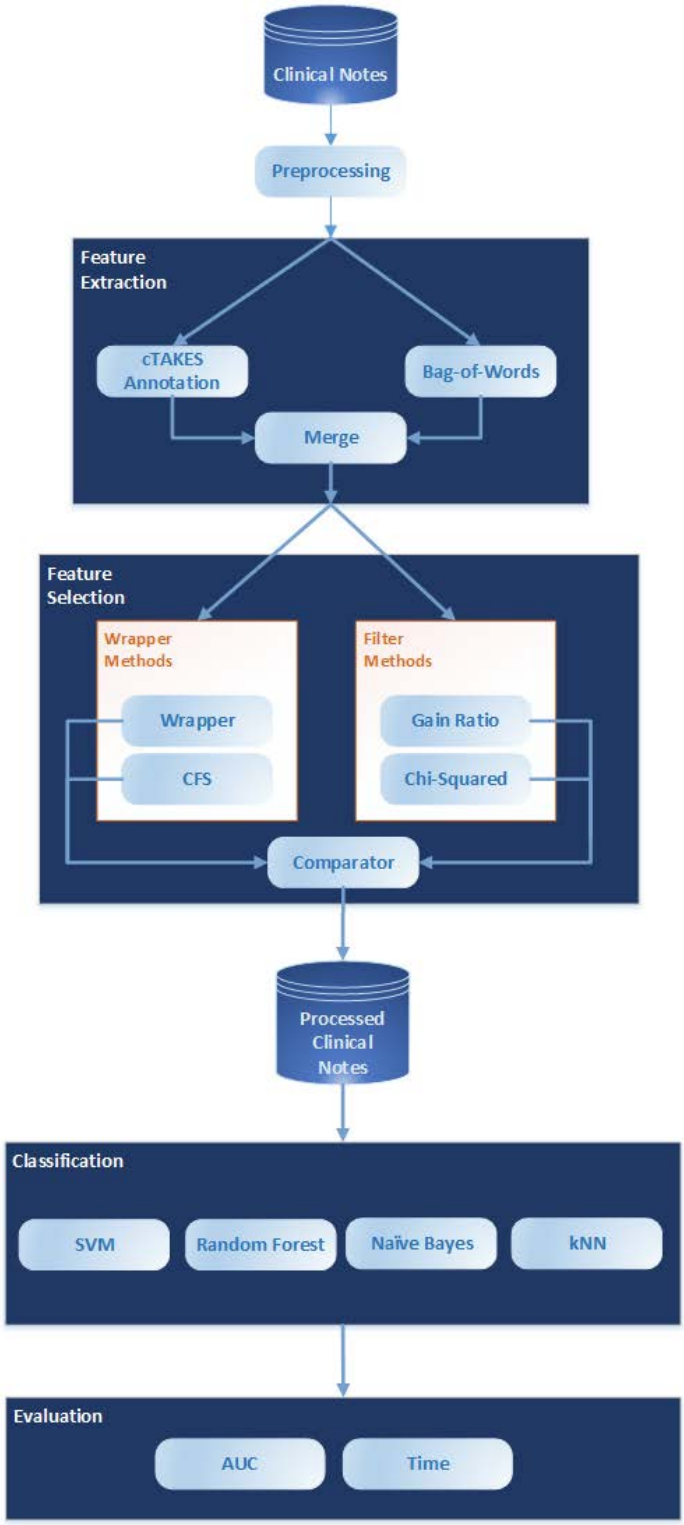

Figure 4.1 Diagram Outlining Proposed NLP PHRS

4.1.2 Dataset

The dataset used for this research consists of 1,248 clinical notes from COPD patients over the period of five years. Patients diagnosed with COPD as a primary or contributing factor are included in this dataset. The number of features extracted for the entire dataset is 5,428 with an average of 593 features per clinical note. cTAKES extracted 1,220 features for the entire dataset and after preprocessing, 4,208 features are represented by bag-of-words. Readmission status is considered true if patient that was readmitted to the same hospital within 30 days of discharge. The class distribution for this dataset is 14.3% positive and 85.7% negative. The data evaluated using n-fold cross validation where n=10.

4.2 Cost

Although unplanned hospital readmission has been studied for decades, recent federal legislation has caused an increase in research. The Hospital Readmission and Reduction Program (HRRP) has given hospitals financial incentive to lower unplanned hospital readmission. Hospitals are given target readmission rates and formulas dictate the penalties incurred when rates are exceeded. Even though HRRP has driven the increase of PHRS research, the penalties have largely been ignored by existing systems. The cost of readmission is often assumed to be the same as non-readmission. Formulas published by CMS dictate this is clearly not the case. Additionally, hospitals achieving lower readmission rates than CMS targets are not given additional funds. Only penalties are possible. Current systems blindly classify patients without regard to cost, using resources that could potentially be utilized elsewhere.

Cost sensitive learning and classification has been studied extensively and may be

applied to this problem. However, misclassification cost based on formulas and rates offer unique challenges as existing methods often assume fixed misclassification cost. For example, cost sensitive classification may be used by the financial industry to predict which customers should receive loans. Issuing a loan to a customer whom will ultimately default may cost five times as much as the lost interest payments if the customer does indeed repay the loan. This ratio may hold regardless of the previous number of loan acceptances or denials. However, CMS financial penalties are based on a rate and the same patient readmission may cost differing amounts from day-to-day.

This work attempts to overcome those challenges and offer a metric that is more meaningful to hospital staff. C-statistic and AUC are the most often used evaluation metric for PHRS, as evidenced by Kansagara et al. However, this metric may be unclear and be of little use to hospital administrators whom must ultimately translate such metrics into cost and resource allocation. Directly reporting cost may offer a clearer metric rather than relying on a proxy measure of quality. Additionally, little work has been done to show in practice that a model with a high AUC has a low cost. Two models with similar AUC may potentially report different cost due to differing misclassification costs. This work attempts to address these problems in an effort to increase practical adoption of PHRS in the clinical setting.

CMS has published formulas based on HRRP which dictates excessive readmissions on a per-hospital basis. There are two components to this penalty, the base operating Diagnostic Related Group (DRG) payment amount and the Excess Readmission Ratio (ERR). The base operating DRG payment is calculated by CMS using many criteria, including case mix index, labor share, wage index, non-labor share, cost of living

adjustments, technology payments, and total number of Medicare cases [6]. Many of these variables are beyond the control of the medical facility and for modeling purposes DRG cost is assumed to be uncontrollable. The second component to readmission penalty is the ERR. ERR is defined as

$$ERR = \frac{Predicted\ readmissions\ rate}{Expected\ readmissions\ rate} - 1 \quad \textbf{(4.1)}$$

Expected readmission rate is defined as the expected rate of readmission given the hospital's patient demographics and is calculated by CMS using regression models based on national readmission statistics [5]. Predicted readmission rate incorporates the hospital's risk adjusted readmission statistics as to minimize uncontrollable risk factors. Expected readmission rate can be treated as uncontrollable since hospitals have little control over their patient population and demographics. Predicted readmissions rate may be improved by hospitals through the reduction of readmissions. The final equation for 30-day readmission penalty for COPD is defined by CMS as

$$cost = DRG * ERR \quad \textbf{(4.2)}$$

where DRG is the sum of payments made for the DRG and ERR is the Excess Readmission Ratio for patients within the DRG.

The primary performance metric in many reviewed readmissions systems is the c-statistic (also known as AUC) [18]. However, a potentially small number of patients may need successful intervention to meet target rates and analysis of c-statistic alone may produce an incomplete model. Cost and probability of readmission for individual instances needs to be considered. A model which produces a large c-statistic may not produce the lowest cost.

4.2.1 Cost Sensitive Modeling and Evaluation

Many readmission systems based on predictive modeling ignore misclassification cost and model False Positive (FP) and False Negative (FN) misclassifications to have equal cost. This assumption is often untrue. Cost sensitive data mining is a method which assigns cost to misclassification. Table 4.1 shows a confusion matrix. Table 4.2 outlines a cost-sensitive confusion matrix where μ is defined as the cost of a single FP and λ is defined as the cost of a single FN.

	Predicted Positive	**Predicted Negative**
Actual Positive	True Positive	False Negative
Actual Negative	False Positive	True Negative

Table 4.1 Confusion Matrix

	Predicted Positive	**Predicted Negative**
Actual Positive	0	λ
Actual Negative	μ	0

Table 4.2 Cost-Sensitive Confusion Matrix

Integrating cost into predictive models can be categorized into three approaches: (1) Cost-sensitive learning (2) Cost-sensitive classification and (3) Cost-sensitive evaluation.

Cost-sensitive learning builds a classifier using misclassification cost parameters. The algorithm for building the classifier often times must be modified and may not be practical. Additionally, since the model has been built using a cost matrix, the cost cannot

be easily updated when classifying new instances. For instances where cost is a constant relationship, such as FN costing twice as much as FP, or a constant monetary value, modifying the model is not necessary. However, if cost is based on a rate which incorporates elements of the cost matrix, the model must be updated as new test instances become available. Though the ground-truth label may not be known during classification, a probability of classification may be assigned and this information immediately incorporated into the cost matrix for new instances. The updated cost matrix may result in a more accurate representation of the actual cost.

Cost-sensitive classification builds a classifier with no additional cost parameters. Cost is incorporated when classifying new instances. Posterior probabilities are used to determine the instance's class using the cost matrix. The advantage to this method is there is no need to modify the algorithm building the classifier. However, cost-sensitive classification requires the classifier have the ability to produce posterior probabilities. Some classifiers, such as Naïve Bayes, can produce posterior probabilities with no additional work as they are fundamental to building the classifier. Other classifiers, such as decision trees, must use additional techniques to coerce a posterior probability. Weka, a popular data mining software tool, requires classifier implementations have the ability to produce posterior probabilities through the *distributionForInstance* function. Therefore, no additional work needs to be done to support cost-sensitive classification when using this tool.

$$class = \begin{cases} positive, & if\ p_+ > \dfrac{\mu}{\lambda + \mu} \\ negative, & otherwise \end{cases} \qquad \textbf{(4.3)}$$

Cost sensitive classification is defined by eq. (4.3), where p_+ is probability of an

instance (determined by the classifier) being true and $\frac{\mu}{\lambda+\mu}$ represents the threshold of classification. When the threshold of classification is reached, the instance is classified as positive.

Cost sensitive evaluation uses the cost matrix to determine the cost of misclassification. The total cost of misclassification is often obtained by the summation of all instance costs. An average cost per instance may be desirable and is often calculated when reporting cost sensitive evaluation. In the case where misclassification cost is not constant, summation of all instance costs may not be sufficient.

4.2.2 Classification

The classification algorithms chosen for this research are Naïve Bayes (NB), Random Forest (RF), Support Vector Machines (SVM), k-Nearest Neighbors (kNN), C4.5, Bagging with REPTree, and Boosting with Decision Stump. The data mining toolkit used in this research is Weka 3.6.14. The Area Under ROC Curve (AUC) is used as the primary performance metric.

4.2.3 Cost Evaluation

C	Total cost of Diagnostic Related Group (DRG)
C_{np}	Total cost of DRG for new patient(s) under analysis
ω	Risk adjustment factor
R	Number of readmissions for current fiscal period
R_{fn}	Number of FN readmissions in new patient analysis
P	Number of total patients in DRG for current fiscal period
ρ	Expected rate
$\hat{\rho}$	Predicted rate
p_r	Probability of needing readmission
N	Number of new patients under consideration
N_s	Number of new patients to select for intervention
p_s	Probability of intervention success
$\bar{c_t}$	Average cost of patient intervention (i.e. home healthcare professional)

Table 4.3 Model Variables

CMS calculates penalties based on eq. (4.2) and is the foundation of our cost evaluation model. Table 4.3 outlines definitions for variables used in our cost sensitive model. As previously stated, the individual components of the DRG amount is outside the scope of our model and treated as a single cost variable. Many components of this variable are difficult or impractical to influence administratively. Expected rate is calculated based on national readmission statistics and is also treated as a single variable with which the facility has no control. However, the predicted rate can be improved. CMS uses regression modeling to determine the number of readmissions with which a hospital is held responsible. Rewriting eq. (4.2), ERR can be expanded and risk adjustment factored out. Risk adjustment is the fraction of actual readmissions between [0,1] for which a hospital is responsible.

$$cost_{cms} = C * \left(\frac{\omega\hat{\rho}}{\rho} - 1\right) \quad \textbf{(4.4)}$$

Each hospital has a set of patients for which the ground truth readmissions status is known. This is due either to 30-days having lapsed or the patient having experienced readmission. These patients are denoted P and R respectively. By definition, $\hat{\rho} = \frac{R}{P}$ and can be expanded in our equation.

$$cost_{base} = (C)\left(\frac{\omega\left(\frac{R}{P}\right)}{\rho} - 1\right) \quad \textbf{(4.5)}$$

When a new patient is added to our model and the ground truth readmission status is known to be a readmission, R increases by 1, P increases by 1, and the DRG increases

by the cost of that patient. The expected rate ρ remains unaffected as this is set by CMS. λ is defined as the cost the hospital will incur due to increased readmission rates. The misclassification cost for a single isolated FN instance is defined in eq. (4.6).

$$\lambda = \left(C + C_{np}\right)\left(\frac{\omega\left(\frac{R+1}{P+1}\right)}{\rho} - 1\right) - cost_{cms} \tag{4.6}$$

In the case of cost sensitive classification for hospital readmission, we can define μ as the cost of intervention. Intervention is often in the form of a home health care nurse. The patient would not have been readmitted to the hospital, therefore any investment towards preventing readmission is lost.

The total cost of FN misclassification is not the sum of all individual FN misclassifications, however. CMS calculates cost using ERR, as defined in eq. (4.1). The increase in readmissions are the patients which needed but did not receive intervention. The increase in the total number of patients are the number of patients evaluated. The difference between total FN evaluator misclassification performance and the currently calculated CMS cost is represented by λ_{total}. Eq. (4.7) represents this total cost. The total misclassification cost for new patients under analysis is the sum of λ_{total} and μ_{total} as shown in eq. (4.9).

$$\lambda_{total} = (C + C_{np})\left(\frac{\omega\left(\frac{R+R_{fn}}{P+N}\right)}{\rho} - 1\right) - cost_{cms} \tag{4.7}$$

$$\mu_{total} = \sum \mu \tag{4.8}$$

$$cost_{total} = \lambda_{total} + \mu_{total} \tag{4.9}$$

4.2.4 Cost-Sensitive Classification

Patient readmission prediction can occur during two time periods: (1) During the discharge process and (2) Nightly batch processing. Predicting patient readmission probability during discharge is advantageous in that the patient is still in close contact with the facility. Intervention can be setup via scheduling of a home health care nurse or other medical professional. When a patient has left the facility, contacting them again may be difficult. Cost-sensitive classification for any single instance in isolation is determined by eq. (4.6).

Nightly batch processing loses the advantage of immediate patient contact, but gains the advantage of a non-fixed FN cost. Since a batch of instances are available, λ can be updated during each classification iteration. For each iteration, the number of patients increases by one, but the number of readmissions increases by the probability of the current patient resulting in readmission. Eq. (4.10). defines an updatable λ, which may more accurately represent cost over the previously proposed method.

$$\lambda_n = \left(C + C_{np}\right)\left(\frac{\omega\left(\frac{R + \sum_{i=1}^{n} p_r}{P + n}\right)}{\rho} - 1\right) - \sum_{i=0}^{n-1} \lambda_i \tag{4.10}$$

$$\lambda_0 = cost_{cms} \tag{4.11}$$

4.2.5 Example Calculation

Given the following starting assumptions: C is $10,000,000; ω is 1; R is 202; P is 1,000; ρ is .200; $\hat{\rho}$ is .202, first, the base ERR and cost must be calculated.

$$ERR_{cms} = \frac{\omega\hat{\rho}}{\rho} - 1 = \frac{.202}{.200} - 1 = 0.01$$

$$cost_{cms} = C * \left(\frac{\omega\hat{\rho}}{\rho} - 1\right)$$

$$= C * ERR_{cms}$$

$$= \$10{,}000{,}000 * 0.01 = \$100{,}000$$

This represents the current cost in CMS penalties. These are patients for which ground truth label can be applied as the 30-day readmission window has passed. At this point, the classification status of these patients cannot be affected through intervention or otherwise.

If a single new patient were presented and patient is a readmission, the cost of the single patient is as follows. The cost of each patient treatment for this DRG is assumed to be C_{np} = \$10,000.

$$ERR_1 = \frac{\omega\left(\frac{R+1}{P+1}\right)}{\rho} - 1 = \frac{\left(\frac{202+1}{1000+1}\right)}{.200} - 1 = 0.013986$$

$$\lambda_1 = \left(C + C_{np}\right)(ERR_1) - cost_{cms}$$

$$= (\$10{,}010{,}000)(0.01398) - \$100{,}000 = \$40{,}000$$

The cost of this patient arriving and later needing readmission is \$40,000. When analyzed in isolation, the cost of a single readmission will cost this amount. However, when calculating the subsequent cost of additional readmissions, the ground truth label of this readmission may not be known for up to 30 days. A more accurate representation of the potential CMS penalty would use Eq. (4.10). Using predictive modeling, a probability of readmission can be assigned until the ground truth label is known for this instance. This may lead to a closer cost estimate rather than simply assuming all new readmission instances to cost \$40,000. Assuming the instance was classified with $p_r = 0.6$ and n = 1 new instances, the previous example could then be rewritten as the following.

$$ERR_1 = \frac{\omega\left(\frac{R + p_r}{P + n}\right)}{\rho} - 1 = \frac{\left(\frac{202 + .6}{1000 + 1}\right)}{.200} - 1 = 0.01198$$

$$\lambda_1 = \left(C + C_{np}\right)ERR_1 - cost_{cms}$$

$$= (\$10{,}010{,}000)(0.01198) - \$100{,}000 = \$20{,}000$$

When the next patient is ready have misclassification cost calculated, it is assumed we do not know the ground truth classification of the first patient. Assuming the model reported $p_r = 0.8$ for the next patient, $\sum_{i=1}^{n} p_r = 1.4$ and n = 2.

$$ERR_2 = \frac{\omega\left(\frac{R + \sum_{i=1}^{n} p_r}{P + n}\right)}{\rho} - 1 = \frac{\left(\frac{202 + 1.4}{1000 + 2}\right)}{.200} - 1 = 0.01497$$

$$\lambda_2 = \left(C + C_{np}\right)ERR_2 - (\lambda_1 + cost_{cms})$$

$$= (\$10{,}020{,}000)(0.01497) - (\$120{,}000) = \$30{,}000$$

When the third patient is ready have misclassification cost calculated, assume the readmission model reported $p_r = 0.9$, $\sum_{i=1}^{n} p_r = 2.3$, and n = 3.

$$ERR_3 = \frac{\omega\left(\frac{R + \sum_{i=1}^{n} p_r}{P + n}\right)}{\rho} - 1 = \frac{\left(\frac{202 + 2.3}{1000 + 3}\right)}{.200} - 1 = 0.01844$$

$$\lambda_3 = \left(C + C_{np}\right)ERR_3 - (\lambda_2 + \lambda_1 + cost_{cms})$$

$$= (\$10{,}030{,}000)(0.01844) - (\$150{,}000) = \$35{,}000$$

As ground truth labels become available, these calculations may be updated with different starting assumptions to reflect the new information that has become available.

4.2.6 Cost Reduction

Given a set of *N* patients, it is desirable to choose the smallest subset for which to intervene and send a home healthcare professional. This analysis can often be done as a nightly batch process in order to gather a sufficiently large number of patients for analysis.

Before processing begins, patients must have readmission probability assigned using our model, then sorted by probability readmission. Patients with the highest probability of readmission are analyzed first.

The net cost of patient intervention must include the possibility that intervention may not work. Internal statistics regarding the effectiveness of patient intervention may be collected and is represented by p_s. Assuming N patients, of which we intervene for N_s, we can define ERR as follows

$$ERR_{N_s} = \frac{\omega\left(\frac{R + (1 - p_s)N_s + \sum_{i=N_s}^{N} p_{r_i}}{P + N}\right)}{\rho} - 1 \qquad \textbf{(4.12)}$$

Where $(1 - p_s)N_s$ represents the probable number of readmissions that will still occur even though we have intervened. N_s can be increased iteratively until either ERR $\leq$ 0 or there are no additional patients to analyze. If ERR $\leq$ 0, there are no additional CMS penalties and resources for preventing readmission can be diverted elsewhere. Figure 4.2 describes the process for finding the optimal number of patients in which to select for intervention.

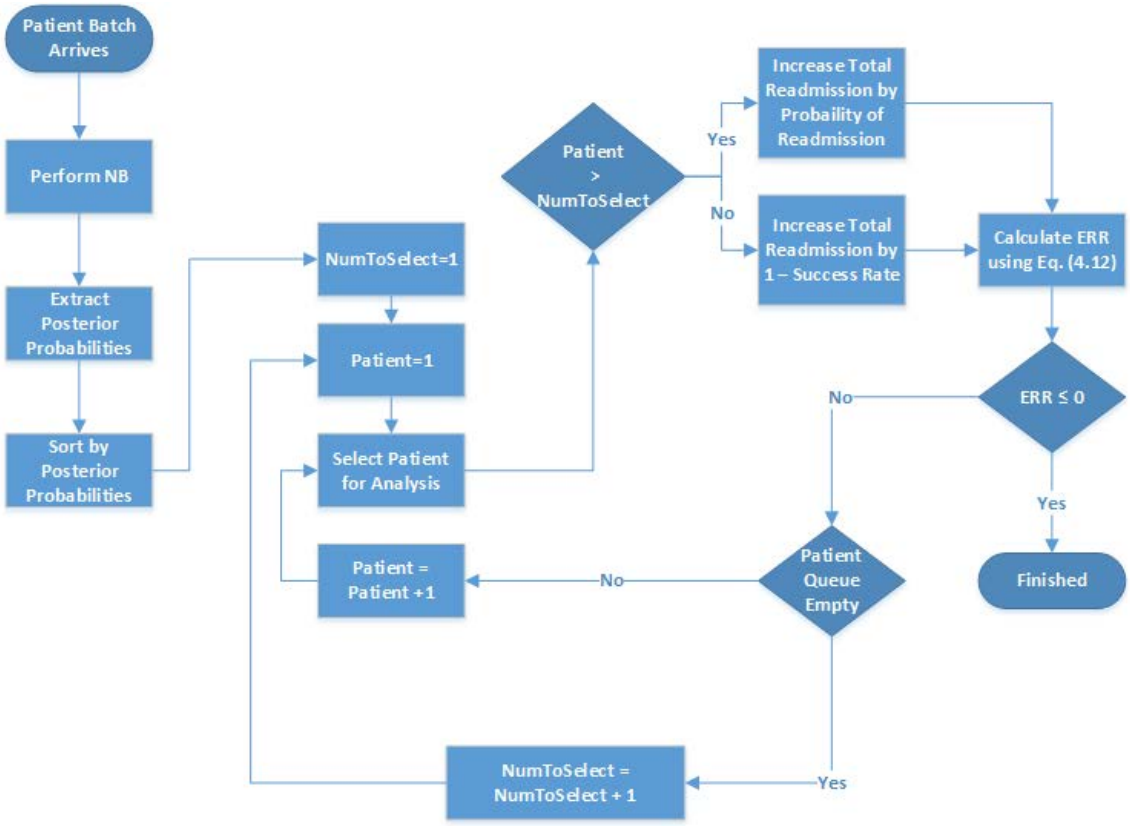

Figure 4.2 Flowchart of Selecting Optimal Number of Patients for Post-Discharge Care

4.2.7 Optimal cost

A more complete model would incorporate the total cost of intervention which includes the average cost of a home healthcare professional, represented as $\bar{c_t}$. Assuming N_s interventions, the current total cost of intervention will be the following.

$$cost_{N_s} = (C + C_{np}) * \left(ERR_{N_s}\right) + \bar{c_t} N_s \quad \textbf{(4.13)}$$

This represents the total cost of intervention given N patients and N_s interventions. Since the patients have been sorted by decreasing order of readmission probability and number of patients in analysis is constant, iteratively increasing the number of patients to intervene until a minimum cost is achieved is possible. When $ERR_{N_s} \leq 0$, no additional cost savings are possible as CMS does not refund medical facilities for exceeding expected

rates. No penalties will be incurred, but cost of sending a home healthcare professional remains. If the cost of $\bar{c_t}$ is very high or set of patients very unlikely to require readmission, a minimum cost where $ERR_{N_s} > 0$ is possible, but rare. This scenario indicates that the patients are very unlikely to need readmission and cost of intervention high enough that it is less expensive to pay CMS penalties than to intervene. Due to the generally high cost of HRRP penalties, this scenario is rare, but possible.

4.2.8 Dataset

The dataset chosen for experimentation contains 1,248 hospital discharge summaries containing COPD as a primary or secondary diagnosis. Features are extracted using a bag-of-words representation and a total of 5,429 features exist. The class distribution is 14.32% readmission and 85.68% non-readmission. Stratified 10-fold cross validation is used in evaluating performance. Single reported values represent the mean value of 10-folds.

4.3 Latent Topic Ensemble Learning

Obtaining useful data is often a difficult challenge in the clinical domain. Clinical data contains temporal issues which may cause data to lose predictive power over time. For example, a model created during flu season may be less useful during the Summer and contain bias toward certain features. Sharing recent data among hospitals is a potential solution, however in practice causes decreased PHRS performance. Patient demographics and disease distribution can differ vastly and there has been little published success when combining data from multiple hospitals [88].

Figure 4.3 shows the vast difference in feature distributions for a large primary hospital plotted against a combined auxiliary dataset of 15 other available hospitals. A line

overlain with an origin of (0,0) and slope=1 shows the ideal distribution of features. A primary and auxiliary dataset whose feature distribution closely follows this line could be considered for a naively merged dataset with little risk of decreased model performance. However, as shown in Figure 4.3, this is clearly not the case.

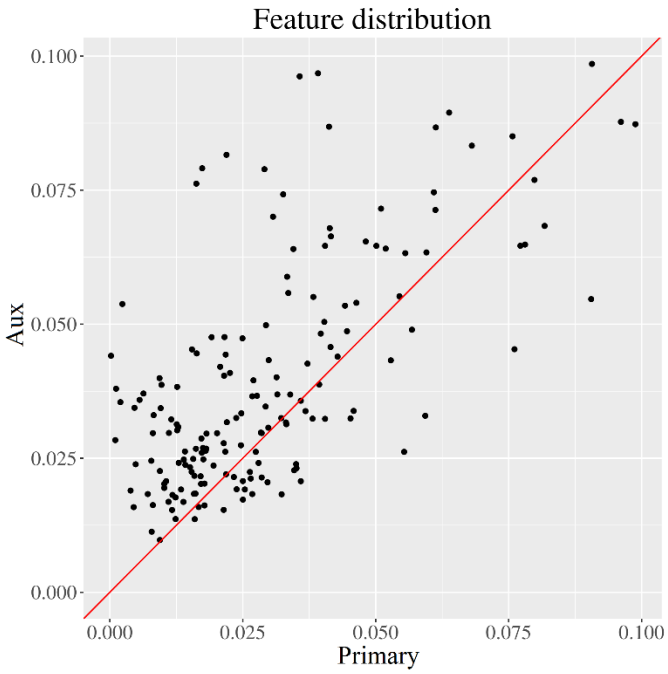

Figure 4.3 Scatterplot of Feature Value Distribution for a Primary Hospital vs All Available Source Hospitals Combined

Many localized models only use a small portion of available data. For example, a model may be built using only instances that contain COPD as a primary diagnosis (as shown in Figure 4.4). Such models result in better performance because diabetes patient's class distributions are better aligned than patients of differing diseases between hospitals. Creating a disease specific model may result in better performance at the cost of discarding many potentially useful instances. Additionally, patients may suffer from multiple diseases to varying degrees. Current models often assign the patient a primary diagnosis and use a single model for readmission classification. However, patients may belong to several

models of varying degrees, requiring an ensemble of models to fully represent the patient.

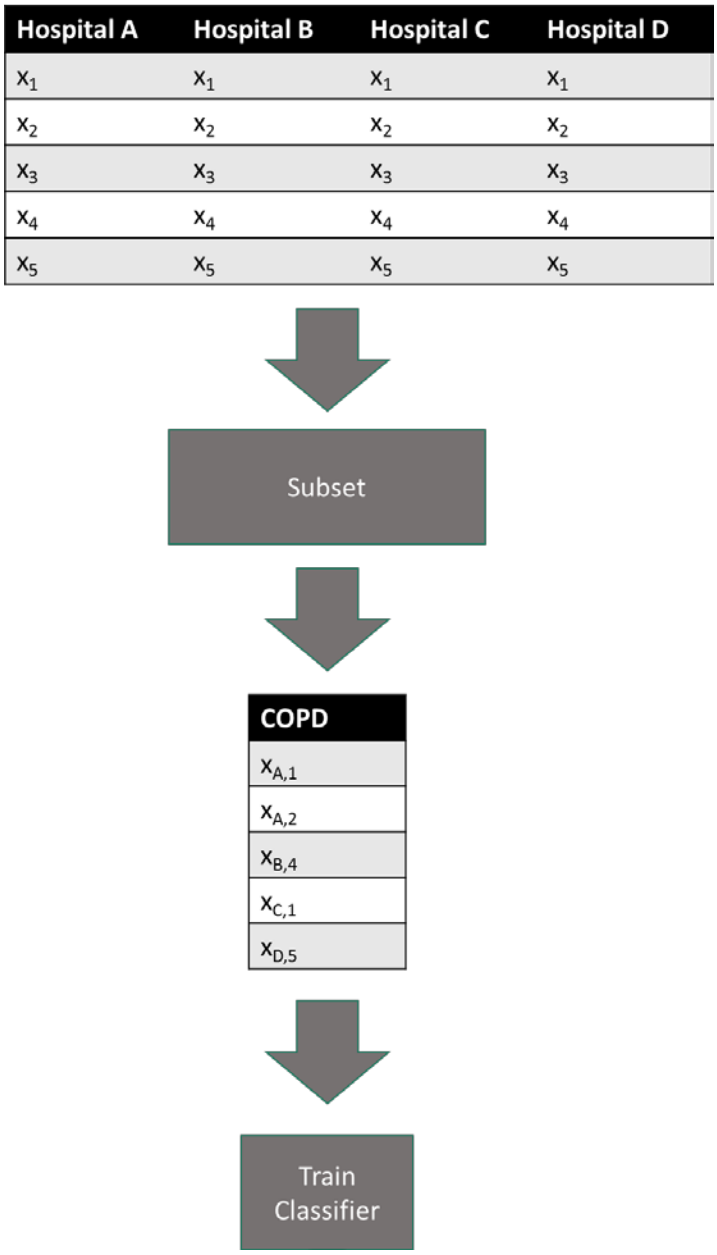

Figure 4.4 Common Method for Subsetting Data When Creating PHRS

While a good predictive system should have the ability to intelligently incorporate all available data into model creation and improve overall performance, no existing hospital readmission system has shown statistically significant performance improvements when attempting to incorporate all available auxiliary data. In this work, Latent Dirichlet

Allocation (LDA) topic modeling is used to find related groups of patients. For example, topic modeling may discover a group of patients suffering from substance abuse. Additionally, patients often belong to several groups and LDA assigns each instance a group membership weight. This allows all available data to be used to create many models.

More specifically, we represent clinical notes as a *bag-of-words* for predictive modeling. This representation is generally highly dimensional, so we use topic modeling, LDA [89], to handle high dimensional medical notes and allow data from multiple sources to be carefully combined for readmission predictive modeling.

Latent Topic Ensemble Learning (LTEL) attempts to address all of these shortcomings. This system uses unstructured clinical notes with LDA to extract topics and assign instances to those topics. A single instance may belong to multiple topics and membership is represented as a weighted value. Models are created per-topic rather than per-hospital and new instances classified using a weighted membership ensemble with soft majority voting. Evaluation is performed using CMS cost formulas. LTEL represents the potentially first published system which addresses all of these concerns successfully.

4.3.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is an unsupervised machine learning algorithm which attempts to group a set of training instances into topics. LDA is often used in the field of text mining and can offer insights into the structure and relationships of data. A predefined number of topics must be provided as a parameter to LDA and often requires experimentation as to the number of optimal topics for a given dataset. Additionally, topics may be difficult to interpret because latent relationships exist and require a domain expert to assign semantic meaning. LDA has been used extensively in the field of text mining, but

has seen little use in text mining of clinical notes and no published work as of yet has used LDA as a method for improving hospital readmission model performance.

4.3.2 Auxiliary Data

Often times hospital readmission data from multiple medical facilities is available when creating a new readmission model. Intuitively, one might attempt to combine all available data into a single classifier. However, classification distribution, patient population, and data entry methods vary amongst hospitals. Simply naively combining all available instances will often result in a model with poor results.

The primary hospital is the hospital which contains instances for which classification is desired. Auxiliary data from available hospitals contains instances which we would like to use to strengthen the performance of the primary hospital. The primary and auxiliary hospitals often have different classification distributions and distribution among features. An algorithm which uses instances from auxiliary hospitals to build a predictive model on the primary hospital must address this or many times the result is a model which performs worse than a model which used only data from the primary hospital. When this occurs, it is known as model degradation.

4.3.3 LTEL

The LTEL algorithm is described below. First, the classifiers are trained (a high level block diagram of this process is shown in Figure 4.5). Topics are created using the combined primary training and auxiliary instances. Each instance is then assigned a membership weight to each topic. These weights are between [0,1] and sum to 1 for each instance. A classifier is then built using all instances and weights. Instances with a greater weight have a proportionally greater influence over classifier creation. The number of

trained classifiers and topics are equal and have 1-to-1 relationship.

Algorithm 1 LTEL Model Training Process

procedure LTEL_TRAIN(D_{prm},D_{aux}, k)
▷ D_{prm}: Primary hospital data; D_{aux} data from other hospitals;
▷ k: number of latent topics;
    $D_{train} \leftarrow D_{prm} \cup D_{aux}$ ▷ Combining primary and auxiliary data;
    $T \leftarrow LDA(D_{train}, k)$; ▷ Find latent topics;
    for instance $x_i \in D_{train}$ **do** ▷ Find instance-topic weight;
        $x_i^w \leftarrow Instance\ Topic\ Weight(T, x_i)$
        $D_{train}^w = D_{train}^w \cup x_i^w$
    end for
    for topic $t_j \in T$ **do** ▷ Learn topic specific classifiers;
        $D_{train}^{w_{t_j}} \leftarrow Weighted\ Instances\ to\ Topic\ (D_{train}^w, t_j)$;
        $C^{t_j} \leftarrow Train\ Classifier(D_{train}^{w_{t_j}})$ ▷ Topic specific classifier;
    end for
    return (C, T)
end procedure

When an unseen instance needs classification, topic membership weights for that instance are calculated using the previously discovered topics. Posterior probability is found using the previously trained classifiers, weighted by the instance's membership of that topic. FN misclassification cost (λ) is then calculated using either fixed or updatable cost equations and the instance is then classified using Eq. (4.3). A high level block diagram of this process is shown in Figure 4.6.

Algorithm 2 LTEL classification algorithm

▷ Classify unseen test instances
procedure LTEL_CLASSIFY(D_{test}, C, T, μ)
▷ D_{test}: Primary hospital test data; ▷ C: Trained classifier ensemble;
▷ T: Latent topics; ▷ μ: FP misclassification cost;
    for instance $x_i \in D_{test}$ **do**
        $x_i^w \leftarrow LDA(T, x_i)$ ▷ Find LDA topic weights for instance
        for Topic Classifier $C^{t_j} \in C$ **do**
            $\Sigma_+ \leftarrow x_i^{w_{t_j}} * Posterior(x_i, C^{t_j})$ ▷ Sum of weighted posterior
        end for
        $\lambda \leftarrow \lambda(\ldots)$; ▷ Update λ using Eq. (4.10)
        if $\Sigma_+ > \frac{\mu}{\lambda+\mu}$ **then** ▷ Classify using Eq. (4.3)
            $\hat{y}_i \leftarrow (+)$
        else
            $\hat{y}_i \leftarrow (-)$
        end if
    end for
    return R ▷ Class predictions of D_{test}
end procedure

Performance evaluation compares the ground truth labels to the classification predictions, calculating cost using CMS equations based on whether the classification is correct. As FP and FN have different cost implications, these are tracked separately. The total misclassification cost is then reported.

Algorithm 3 LTEL performance evaluation

▷ Performance evaluation of test instances
procedure LTEL_EVALUATE(D_{test}, R)
▷ D_{test}: Primary hospital test data;
▷ R: Classified results from LTEL_Classify;
    $R_{fp} \leftarrow 0$; $R_{fn} \leftarrow 0$
    $X \leftarrow D_{test} \cup R$ ▷ Merge predictions and labels
    for instance $x_i \in X$ **do**
        ▷ Compare ground truth *vs.* predicted label
        if $\hat{y}_i == (+)$ *and* $y_i == (-)$ **then**
            $R_{fp} \leftarrow R_{fp} + 1$
        else if $\hat{y}_i == (-)$ *and* $y_i == (+)$ **then**
            $R_{fn} \leftarrow R_{fn} + 1$
        end if
    end for
    $\lambda_{total} \leftarrow \lambda(\ldots)$; ▷ Calculate FN cost using Eq. (4.7)
    $\mu_{total} \leftarrow \mu(\ldots)$; ▷ Calculate FP cost using Eq. (4.8)
    return $\lambda_{total} + \mu_{total}$; ▷ Calculate total cost using Eq. (4.9)
end procedure

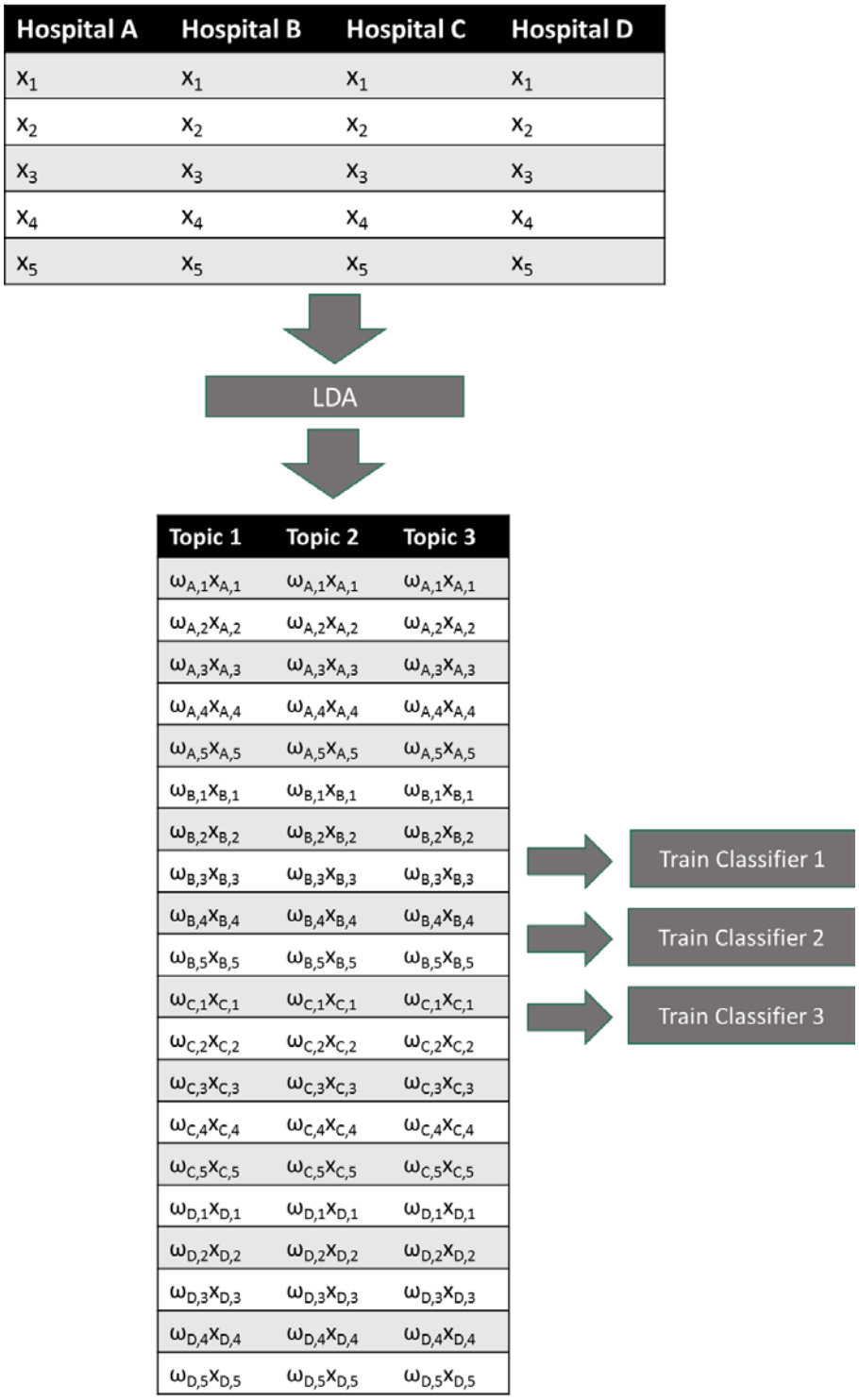

Figure 4.5 Training Phase of LTEL

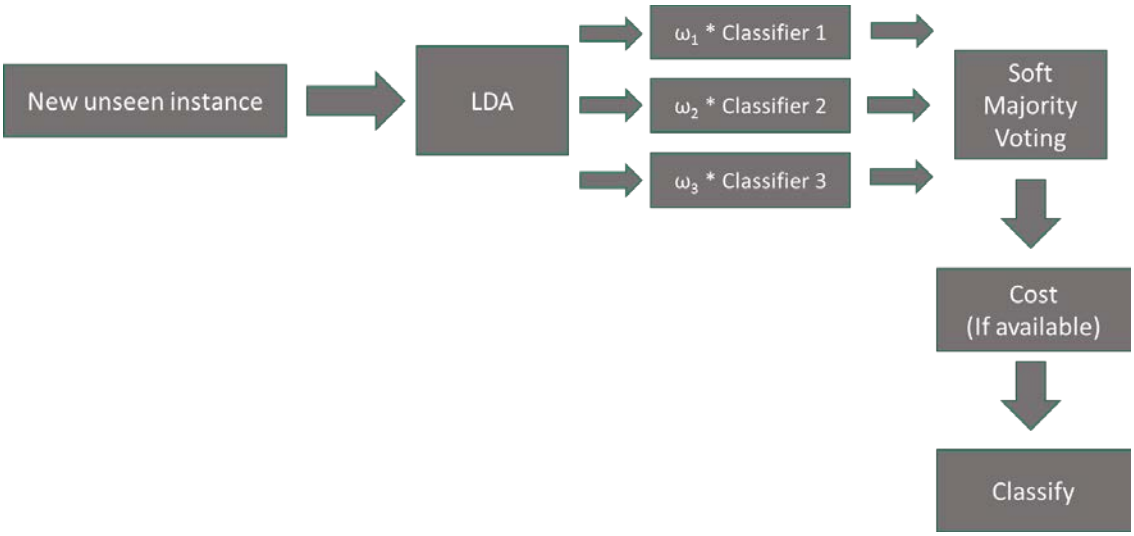

Figure 4.6 Classification Phase of LTEL

4.3.4 Dataset

Hospital	Instances	Readmission Rate
A	15991	0.0958
B	193	0.0000
C	342	0.0146
D	1056	0.0643
E	13589	0.0884
F	82	0.0243
G	1983	0.0927
H	2172	0.0465
I	3767	0.0501
J	8698	0.0756
K	209	0.0095
L	3704	0.0534
M	6790	0.0602
N	2222	0.0567
O	57	0.0175
P	1859	0.0268

Table 4.4 Description of All Hospitals

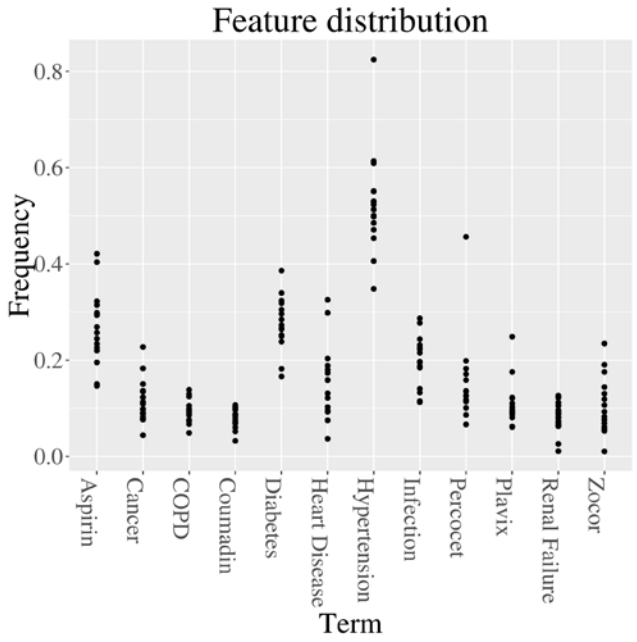

Figure 4.7 Distribution of Common Diseases for Each Hospital

The dataset for this research contains data collected from 16 regional hospitals. Data is split and evaluated using 10-fold cross validation. Hospitals vary greatly in size, patient demographics, and readmission rates as shown in Table 4.4 and Figure 4.7. An additional challenge presented by this dataset is several hospitals only have limited data available. Rather than discard hospitals with limited data, these are included as auxiliary instances when training LTEL. However, the small number of instances may not be practical to evaluate using 10-fold cross validation. Limited data may be due to a hospital being recently opened, changing IT systems, or low volume of patients for a selected time period.

Topic	Important Terms	Description
1	Hypertension; Penicillin; Asthmas; Accident	Combination of diseases, drugs, and terms.
2	Hypertension; Diabetes; Coronary Disease; Plavix Lopressor	Heart disease.
3	COPD; Lung Disease; Albuterol; Advair	Diseases and medications related to chronic lung disease.

4	Hypertension; Heart Attack; Aspirin	Heart attack.
5	Cancer; Liver Diseases; Percocet	Combination of diseases and pain medication.
6	Percocet; Bone Fracture; Arthritis; Vicodin	Bone diseases and pain medications.
7	Heart Fibrillation Congestive Heart Failure Coronary Disease	Heart conditions known to co-occur.
8	Communicable Disease Urinary Tract Infection Vancomycin	Bacterial infections and anti-biotics due to communicable disease.
9	Kidney Disease; Diabetes; Hypertension	Combination of diseases.
10	Alcohol Abuse; Hepatitis; Liver Disease; Drug Habituation	Liver diseases related to drug and alcohol abuse.

Table 4.5 Topics Discovered by LDA for All Hospitals

An analysis of 10 topics using all hospital data is shown in Table 4.5. Terms which contained the highest importance to the topic were extracted. Many topics contained clear conditions for which to apply a topic. Topic 3 contains terms relating to Chronic Obstructive Pulmonary Disease (COPD) and Topic 6 contains terms relating to patients with bone diseases. These are often accompanied by pain medications as they can be very painful. Topic 10 contains not just diseases, but diseases that pertain to a set of lifestyle conditions. Several topics contain an unclear combination of highly prevalent diseases and medications.

4.3.5 Cost

For comparison purposes, all hospitals were given initial starting assumptions. The fixed misclassification cost of a FP was $800. After consulting with domain experts, this was found to be a reasonable average cost for sending a home healthcare professional to a patient's home for several hours and a possible second visit. Expected readmissions rate was set to .14 and current predicted readmissions rate was .143. The fixed FN

misclassification cost was then calculated to be $61,428.60 assuming an average of 1,000 ground truth labels being available.

4.3.6 Baseline Methods

In our experiments, we implemented the following baseline methods for comparisons.

Primary: This method only uses the data from the primary hospital to train classifier for readmission prediction.

Prm+Aux: This method simply aggregates all data from the primary hospital and the auxiliary hospitals to form one dataset, and then train a model from the aggregated dataset.

Bagging: This is a simile ensemble learning approach which treats each hospital separately. A classifier is trained from each single hospital, and all classifiers equally vote to predict a test instance.

TrAdaBoost: This method is a popular transfer learning algorithm addressing differences in distribution [90]. TrAdaBoost is a modification of the AdaBoost algorithm which creates an ensemble of classifiers and uses a weighted voting mechanism to classify instances. AdaBoost works by creating so-called expert classifiers that perform well on a certain subset of training data. Each classifier is weight based upon how many instances that expert can correctly classify. TrAdaBoost iteratively builds an ensemble of classifiers, lowering the weights of diff-distribution instances when incorrectly predicted during the construction phase of the AdaBoost algorithm. The assumption is that these instances are too different from the target domain and offer little usable information.

LTEL: This represents the proposed method which uses latent topic ensemble

learning for hospital readmission prediction.

4.3.7 Feature Extraction

In our experiments, discharge summaries are annotated using Apache cTAKES. Annotations containing diseases & disorders, medications, and anatomical site are used. Annotations are normalized to a UMLS CID to increase the quality of features. The corpus contains 7,112 extracted features and non-cTAKES features are not included. Features are represented using the *bag-of-words* model. Instances with multiple occurrences of the same CID are limited to a single representation thereby limiting each feature to the binary values {0,1}.

4.3.8 Learning Algorithms

The algorithms chosen for this research are Naïve Bayes (NB), k-nearest neighbors (kNN), Linear Regression (LR), and Support Vector Machine (SVM). Decision trees were considered for this research but early experimentation revealed due to high misclassification cost, they had no discriminative ability and would often classify all instances as belonging to the positive class. This, coupled with their computationally intensive nature disqualified their inclusion in this research. Weka 3.6 is used for the implementations of all algorithms.

NB is computationally quick and performs well for highly dimensional data. kNN uses distance functions to find the most similar instances to the instance under consideration. Similar to NB, kNN is computationally fast. LR forms a linear decision boundary between instances. SVM forms a similar linear boundary, but attempts to maximize the margin between classes.

4.4 Predicting Primary Cause of Readmission

An often overlooked component in PHRS is identifying the primary cause of readmission. Identifying potential hospital readmissions may have little value if medical staff are attempting to mitigate readmission under incorrect assumptions. The primary cause of readmission is often not the same as the index admission. A two-stage approach may increase the quality of information available to medical professionals. First, the patient is classified for readmission using binary classification methods. Then, if the patient is classified as a potential readmission, the primary cause of that readmission is predicted using multi-class classification. Although many systems attempt to increase model transparency using methods such as risk scores and decision trees, there exists little research into multi-class classification for hospital readmissions.

Obtaining clinical data for research purposes is often difficult. Laws protecting patient Personally Identifiable Information (PII) have existed for more than twenty years [91]. Many organizations only allow data access to researchers employed at the organization while under close supervision by review boards. A side-effect of this need for privacy has caused data driven readmission research to have low rates of research reproduction and validation. Additionally, systems which attempt to computationally analyze data are often difficult or impossible to compare as they are built from vastly different data. This is atypical of data driven research as other domains have standard datasets from which to compare methodologies. These datasets include the Reuters text categorization dataset [92] and Amazon reviews dataset [93].

Recently, a large high-quality dataset of patient readmissions was released by the federal government [94]. This dataset may potentially increase direct comparison of

methods. However, few researchers have published systems based on this dataset. A key feature included in this dataset is the primary cause of readmission, which is not tracked in many datasets.

Identifying the primary cause of hospital readmissions remains a relatively unexplored topic. The HCUP NRD dataset is chosen as the primary dataset for this research. The two driving factors for this decision are (1) the lack of required administrative codes to determine readmission cause in the unstructured dataset and (2) the need to provide a set of benchmarks and best practices for using this dataset in a PHRS.

4.4.1 Dataset

The Healthcare Cost and Utilization Project (HCUP) was created in the late 1980's in an effort to make available a large comprehensive dataset to enable data driven research [95]. HCUP includes the National Inpatient Sample (NIS) database which collects a vast array of data from patient hospitalizations. The dataset is unique in that it does not only make available aggregate and summary information, but provides data at the individual patient level. NIS is available from the Agency for Healthcare Research and Quality (AHRQ) for a relatively small fee and by attending an online data responsibility and usage training course. Until recently, hospital readmission status was not tracked.

The Nationwide Readmissions Database (NRD) is an extension of NIS which contains readmission status [94]. The availability of a standardized dataset may prove invaluable to data driven readmission prevention systems. The dataset contains national data from years 2013 and 2014. Each year contains approximately 15 million discharge records. In addition to potentially representing a standard dataset for which to compare methodologies, NRD is vastly larger than most datasets procured from a single hospital or

hospital system. Although the NRD may potentially yield significant advances in comparison of readmission prediction methodologies, the NRD has seen little published research in this area as of yet. Additionally, few best practices have emerged for transforming this data into manageable forms for processing. Our research attempts to establish a set of best practices and serve as a baseline for further research which attempts to propose improvements.

Patients admitted to a hospital are assigned a Diagnosis Related Group (DRG). These groups are derived from ICD-9-cm codes and represent the primary reason for admission. Although DRG is useful, many DRG codes exist and can be grouped into 25 higher level categories known as Major Diagnostic Categories (MDC). These categories and descriptions are noted in Table 4.6. Index admission and readmission are often for differing reasons. The MDC can provide a metric for which to measure these differences. As noted in Table 4.6, the rate of same MDC readmission may vastly differ. MDC 24 only encounters 0.049 same MDC readmission, while MDC 14 encounters 0.950 same MDC readmission. Medical professionals may choose to provide targeted mitigation efforts based upon readmission MDC. However, only knowing that patient will likely be readmitted for a different MDC is insufficient and predictive modeling may offer insight to readmission MDC prediction.

MDC	**Description**	**Overall Percentage**	**Percentage of Total Readmissions**	**Percentage of Readmissions with Same MDC**
0	Pre-MDC	0.002	0.002	0.271
1	Nervous System	0.063	0.062	0.417
2	Eye	0.001	0.000	0.192
3	Ear, Nose, Mouth And Throat	0.009	0.007	0.182
4	Respiratory	0.095	0.117	0.484

	System			
5	Circulatory System	0.133	0.169	0.506
6	Digestive System	0.093	0.111	0.459
7	Hepatobiliary System And Pancreas	0.032	0.044	0.479
8	Musculoskeletal System And Connective Tissue	0.101	0.064	0.296
9	Skin, Subcutaneous Tissue And Breast	0.025	0.023	0.285
10	Endocrine, Nutritional And Metabolic System	0.034	0.043	0.306
11	Kidney And Urinary Tract	0.049	0.067	0.305
12	Male Reproductive System	0.004	0.003	0.153
13	Female Reproductive System	0.012	0.007	0.217
14	Pregnancy, Childbirth And Puerperium	0.128	0.039	0.950
15	Newborn And Other Neonates (Perinatal Period)	0.050	0.008	0.778
16	Blood and Blood Forming Organs and Immunological Disorders	0.014	0.027	0.387
17	Myeloproliferative DDs (Poorly Differentiated Neoplasms)	0.008	0.031	0.645
18	Infectious and Parasitic DDs (Systemic or unspecified sites)	0.055	0.061	0.278
19	Mental Diseases	0.043	0.055	0.754

	and Disorders			
20	Alcohol/Drug Use or Induced Mental Disorders	0.013	0.020	0.546
21	Injuries, Poison And Toxic Effect of Drugs	0.014	0.015	0.164
22	Burns	0.0001	0.0006	0.580
23	Factors Influencing Health Status and Other Contacts with Health Services	0.006	0.008	0.109
24	Multiple Significant Trauma	0.002	0.001	0.049
25	Human Immunodeficiency Virus Infection	0.001	0.002	0.540

Table 4.6 Description of MDC Statistics

The HCUP NRD version used in this work is the 2014 HCUP NRD dataset. This dataset contains 14,894,613 instances and 148 features. A selection of features is shown in Table 4.7. 32 features are usable without modification. ICD-9 diagnostic codes are used as the primary clinical data source, represented as DX1 through DX25. Unplanned all-cause 30-day readmission to any hospital is considered a positive hospital readmission. Using this criteria, 0.105 of all instances are readmissions. Stratified cross validation is used to resample the data to create multiple models. 5-fold cross validation is the number of folds created. Weka 3.6 is used to resample the data and the random seed is set to 1.

Feature	**Description**
AGE	Age of the patient
AWEEKEND	Patient was treated during a weekend
DIED	Patient died
DISPUNIFORM	Disposition of the patient at discharge
DMONTH	Discharge month of year
DQTR	Discharge quarter of year
DRG	Diagnosis related group

DRGVER	DRG grouper version
DRG_NoPOA	DRG without present on admission flag
DX[1-25]	ICD-9-cm diagnoses
ELECTIVE	Admission was elective
FEMALE	Patient is female
HCUP_ED	Patient required emergency services
HOSP_NRD	Hospital identification code
LOS	Length of stay
MDC	Major diagnosis group
MDC_NoPOA	MRD without present on admission flag
NCHRONIC	Number of chronic conditions
NDX	Number of ICD-9-cm diagnoses
NPR	Number of ICD-9-cm procedures
ORPROC	Major operating room procedure was required
PAY1	Indicator code of payer
PL_NCHS	Urban-rural classification
PR[1-10]	ICD-9-cm procedures
REHABTRANSFER	Patient was transferred for rehab
RESIDENT	Patient is a resident of the state in which they received treatment
SAMEDAYEVENT	Multiple admission/discharge events in same day
SERVICELINE	Hospitalization type
TOTCHG	Total charges in dollars
YEAR	Discharge year
ZIPINC_QRTL	Quartile of patient's household income

Table 4.7 Variables in the HCUP NRD

4.4.2 Classification

NB is used as the primary classifier for this research due to computational constraints. SVM, kNN, and RF were considered for inclusion, however early experimentation proved model creation time to exceed practical considerations. The HPC platform for which this research was performed was unable to produce a model in less than one week. Administrative limits on this platform do not allow jobs to run for longer than one week and are automatically discontinued. Additionally, ensemble methods were utilized using NB as the base classifier. Bagging and LTEL approaches were utilized during binary classification.

Two stages of classification are utilized to predict primary cause of readmission. First, binary classification is performed to predict patients most likely to need readmission. This method is similar to previous classification tasks used in this research. When a patient is classified as a hospital readmission, the second stage performs multi-class classification to determine the MDC for which the patient is likely belong during readmission.

4.4.3 Feature extraction

NRD represents each hospital visit as an instance. Sociodemographic information is represented as features and requires little additional processing. However, ICD-9 codes are presented in a format suitable for humans to read rather than processing by a machine learning algorithm. Columns DX1 through DX25 contain applicable clinical information encoded as ICD-9. For efficient representation of features, these codes must be aligned. For example, an instance might have a code representing asthma as DX1. Another instance may have asthma represented under DX2. Due to this misalignment, machine learning algorithms don't have an accurate representation of the feature.

Several methods exist for efficient representation of ICD-9 codes. A common method is to count code occurrence regardless of placement. For example, if asthma occurs once in DX1 and once in DX2, asthma is counted twice. Each code is counted then sorted. The highest frequency codes are chosen with an arbitrary cutoff, typically less than 100. Each code is then a feature and frequency within the instance its value. This methodology is common but discards potentially useful information. Including all features leads to a phenomenon known as the "curse of dimensionality" and may additionally cause machine learning algorithms to train models prohibitively slow.

4.4.4 Feature Selection

Extracting features using the above methodology but selecting only the most useful features is desired. Not all features contain useful information and selecting a reasonable subset allows models to train faster while potentially addressing the curse of dimensionality. This method is in contrast to selecting the most frequently occurring ICD-9 codes as frequently occurring terms do not necessarily amount to the most useful. Several methodologies for selecting the most useful features exist. Simply selecting all permutations of potential features then building and evaluating a model may find an optimal subset of features. In practice, this is often slow and impractical. Several statistical measures exist which do not require building and evaluating a model for feature selection. These are often more practical as they only require the calculation of a single statistical measure and run considerably faster.

4.4.5 Evaluation

Previous sections outlined a cost methodology and motivation for use as a primary performance metric. Two basic approaches are used. The first method chooses to evaluate patients in isolation. In the clinical setting, this approach represents a patient arriving, receiving treatment, and during discharge being evaluated for readmission probability. The advantage to this method is that the patient has not left the facility and can potentially have a home healthcare professional assigned immediately. Both cost-sensitive and cost-insensitive classification is performed and average cost per instance presented. The FN cost is determined by eq. (4.6). The second approach is nightly batch processing. The cost may be a more realistic representation of the actual financial penalties imposed, however the patient has already left and may be more difficult to contact. Both methods may be suitable

to the clinical setting and results presented.

Although AUC may be considered an inferior metric to cost in the clinical setting, AUC remains a popular performance metric. Therefore, AUC results are presented in addition to cost.

4.5 Co-Occurring Evidence Discovery

Historically, many medical professionals diagnosed COPD as chronic bronchitis or emphysema. More recently, diseases characterized by chronic cough with sputum production and increasing shortness of breath are encompassed in the blanket diagnosis of COPD [96]. This means that COPD often co-occurs with related lung diseases. However, many diseases that co-occur with COPD are not contained within the family of COPD diseases. For example, hypertension often co-occurs with COPD because smoking increases the risk for both diseases [97]. Other diseases such as asthma may also affect the lungs and have a high co-occurrence with COPD. Additionally, many medications not specifically created for COPD treatment are highly correlated with COPD. Aspirin has been shown to help in the treatment of COPD patients and is often prescribed by medical professionals [98].

The discovery of co-occurring diseases, symptoms, and medications is often invaluable to researchers and medical professionals. Developing treatments in isolation without regard to co-occurring diseases may reduce their effectiveness. A medication developed to treat COPD patients which cannot be taken by those with heart disease may exclude patients with hypertension. Indexes exist which measure the likelihood of patient death based upon which diseases are present [99]. Building these indexes using a computational approach by algorithmically discovering co-occurring diseases, symptoms,

and medications would greatly expand their accuracy and coverage.

Currently, no standard set of ground-truth terms exists for evaluating the performance of COPD co-occurrence analysis. The contribution of our work is (1) proposed methodology and manual creation of an expert reviewed dictionary and (2) proposition of new mathematical formulas and big data computational framework for finding COPD related terms. After the ground truth dictionary has been created, it is evaluated using precision and recall against traditional methods for finding disease and term co-occurrence.

Our research attempts to create a computational framework for the discovery of co-occurring diseases, medications, and symptoms in COPD patients. COPD was chosen because it is tangential to many lung diseases. Clinical notes are used as the primary data source due to a potentially high yield of information. Several NLP techniques are employed in this framework in an effort to maximize the information captured within these notes. With the emergence of electronic patient record databases, many large systems containing big data are now available. Examples of these are the Veterans Affairs (VA) hospital system and England's National Health Service (NHS). Our methodology uses a big data approach to finding co-occurring evidence and is validated using a dataset containing approximately 64,000 instances. Due to rarity of access to databases as large as the VA system, this dataset was the largest available to our research group. However, the methodology was designed with big data techniques as the foundation and can be employed by organizations such as the VA hospital system without scalability issues. Although an increasing number of researchers are using NLP with clinical notes as a data source [100], [101] , few have explored COPD clinical notes [48] and there is no documented evidence

in Google Scholar of this methodology applied to big data.

The Apache Hadoop ecosystem is leveraged for COED. Hadoop Distributed Filesystem (HDFS) is used for the storage and distribution of deidentified patient discharge summaries. Apache Spark is utilized for MapReduce operations and the pyspark python interface is used for programming. Documents are represented as Resilient Distributed Datasets (RDD). Apache cTAKES is used for the extraction of medical terms from unstructured clinical notes. cTAKES offers several UIMA pipelines and UMLS fast-dictionary-lookup is used as the primary pipeline. Disease, medication, and symptom annotations are stored, excluding annotations marked “history” and those that have been negated. UMLS Concept IDs (CID) are extracted from each annotation and used as the primary term identifier.

4.5.1 Dataset

The data used for this study is comprised of 64,371 deidentified patient discharge summaries. 0.0894 of these contain COPD as either a primary diagnosis or contributing factor. The average clinical note contains 20.6 disease, symptom, or medication mentions. Discharge summaries span six years of collection. Diseases and medications may have several spelling variations and abbreviations in common usage.

4.5.2 Co-Occurrence Evidence Discovery Framework

A simplified outline of COED is described in Figure 4.8. When a document arrives for annotation, it passes through the Apache cTAKES pipeline. The pipeline begins with generalized NLP tasks such as tokenization before reaching clinical NLP tasks such as UMLS dictionary lookup. After the document is annotated, it is serialized and held until all document annotations for the dataset are complete. COED then runs corpus-at-a-time

processing using the following components.

Aggregator – Gathers annotations into a single data file suitable for processing. In the Hadoop ecosystem, this is a tab separated file with one line per document.

Analyzer – Documents are counted and terms mapped to COPD and non-COPD lookup tables. Each document is considered a COPD document if COPD was the primary diagnosis. Terms contained within the same document are considered to be COPD terms and counts incremented within the lookup table. Documents that do not contain COPD as a primary diagnosis are mapped to the non-COPD lookup table in a similar fashion.

Score – The scoring mechanism then scores each term using equations and parameters outlined in the next section.

Ranker – Scores are then ranked and recombined with UMLS definitions for user accessible output.

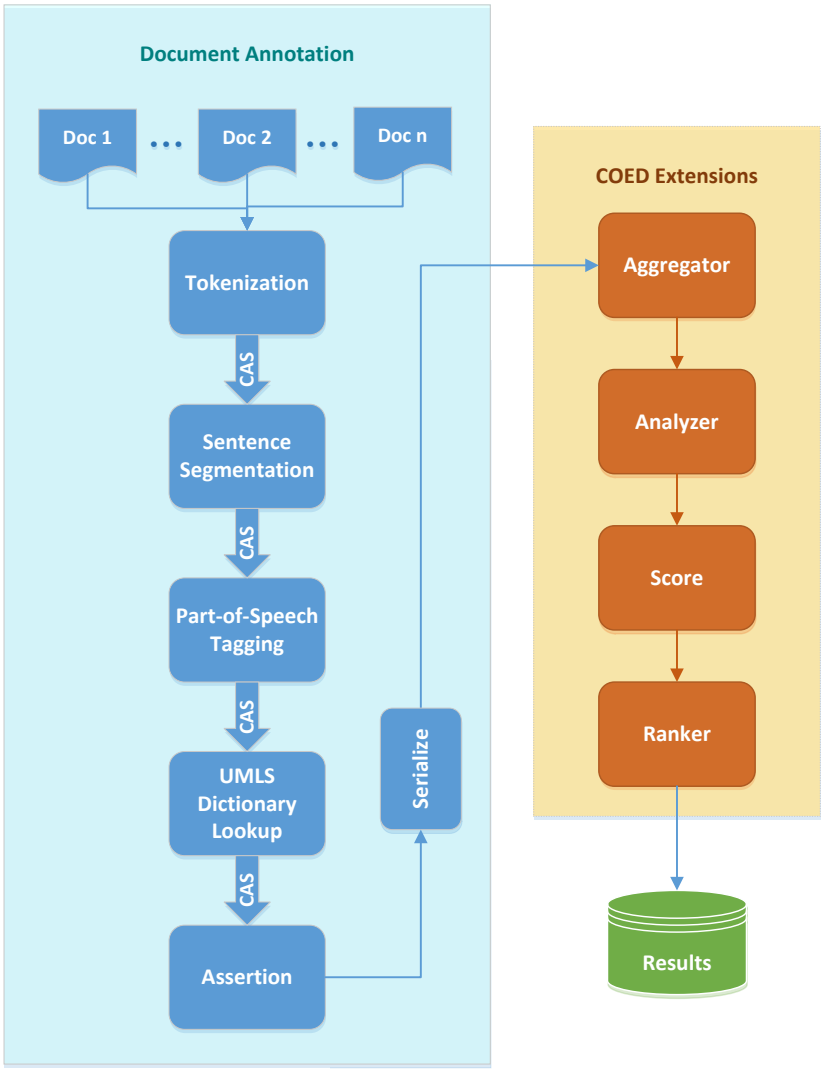

Figure 4.8 A Simplified Outline of COED

A big data prospective of COED is contained in Figure 4.9. Documents are stored in HDFS for annotation. Annotators run independently of each other and may scale to N arbitrary servers, where N is the total number of documents available. Sentence level segmentation may be employed as annotations are performed at the sentence level, increasing the number of servers for which parallelization is available. However, in practice there are typically more documents than servers and is unnecessary. Extensions to cTAKES have been written which allow cTAKES to run as a daemon waiting for a signal

to annotate new documents. This daemon runs on every spark slave instance. A small pyspark wrapper program sends a signal to localhost, thus enabling job control and parallelization to remain in the Hadoop ecosystem.

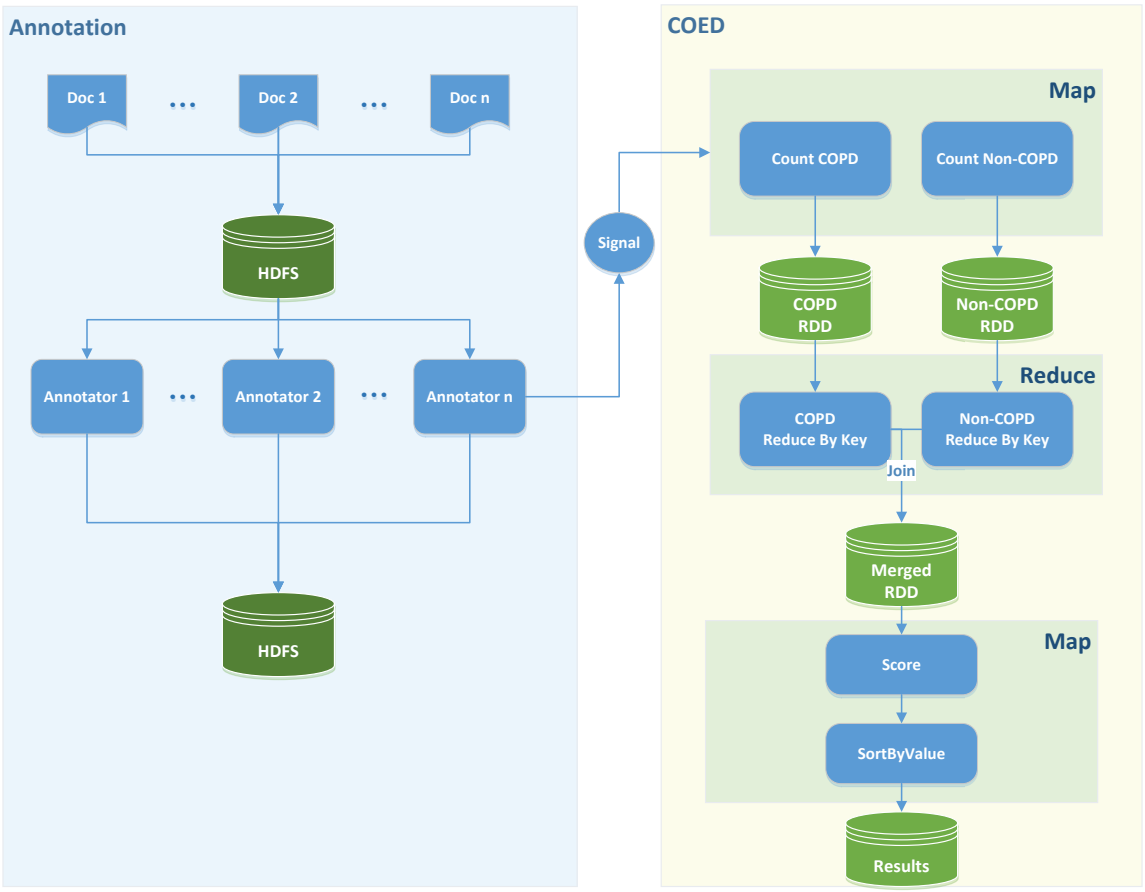

Figure 4.9 Big Data Version of COED as Implemented in the Hadoop Ecosystem Using Apache Spark

After all annotations are complete, annotations are aggregated to a spark compatible file and a signal is sent to the second phase. COED runs as a series of map and reduce tasks. Word counts are performed for COPD and non-COPD documents. Results are stored in two separate RDDs and reduced by key using the *add* callback function. The two RDDs are then joined using the pyspark join() method creating a merged RDD of form (K,(V1,V2)). Each term is then scored using eq. (4.18) and then sorted by value using a

custom sort function. No final reduce operation is required as the previous reduce has ensured distinct keys. Results are then outputted to a file for further analysis.

4.5.3 Score

Co-occurrence of diseases, medications, and symptoms with COPD is traditionally calculated as follows and serves as our baseline co-occurrence method.

$$f_{COPD}(t, D_{COPD}) = \frac{|\{d \in D_{COPD} : t \in d\}|}{|D_{COPD}|} \quad \textbf{(4.14)}$$

Where D_{COPD} is the set of documents containing COPD as a diagnosis and t is the term which co-occurrence is to be calculated. This measurement however, prefers terms which occur frequently in the corpus. For example, research shows arthritis to have a great deal of co-occurrence with COPD [102]. However, arthritis tends to have a great deal of co-occurrence with many diseases as it occurs in 1 in 5 American adults [103]. The primary causes of both diseases are different and risk factors largely independent. Terms which appear often in the document corpus should therefore be penalized, as shown in eq. (4.16).

$$f_{all}(t, D) = \frac{|\{d \in D : t \in d\}|}{|D|} \quad \textbf{(4.15)}$$

$$f(t, D, D_{COPD}) = \frac{f_{COPD}(t, D_{COPD})}{f_{all}(t, D)} \quad \textbf{(4.16)}$$

As the frequency of the term increases in the corpus of documents, the co-occurrence is penalized. This can be helpful in the discovery of terms unique to COPD. However, this will also give a great amount of co-occurrence weight to rare diseases only found in COPD patients. In many cases, a more desirable result would be a lower weighting

of COPD specific terms. Adding a parameter for the penalization of rare terms follows.

$$f(t, D, D_{COPD}) = \frac{f_{COPD}(t, D_{COPD})^{\lambda}}{f_{all}(t, D)} \qquad \textbf{(4.17)}$$

Many variants of this score are possible. The variant primarily used in this research looks at COPD vs non-COPD documents instead of COPD vs all documents. $D_{\overline{COPD}}$ is defined as the set of documents which do not contain COPD as a primary or contributing diagnosis. $\lambda = 2$ is used for experimentation.

$$f(t, D_{\overline{COPD}}, D_{COPD}) = \frac{f_{COPD}(t, D_{COPD})^{\lambda}}{f_{\overline{COPD}}(t, D_{\overline{COPD}})} \qquad \textbf{(4.18)}$$

4.5.4 Evaluation

In order to analyze the performance of retrieved results, a ground truth dictionary of terms was created. 107 diseases, 62 medications, and 46 symptoms were chosen using evidence based approaches. Many criteria were considered when selecting medical terms. Terms which were directly related to COPD such as bronchitis and cough were chosen. Additionally, terms which may not be directly associated with COPD, but have strong common risk factors, such as smoking were chosen. Terms which contain weak associations with common risk factors were not chosen. Smoking is known to exacerbate many diseases such as kidney disease by hardening arteries and reducing blood flow to organs. However, smoking is not the primary cause of kidney disease therefore kidney disease not chosen. Table 4.8 contains a sample of disease, medication, and symptom ground truth terms.

Disease/Disorders	**Symptoms**	**Medications**
Chronic lung disease	Distressed breathing	Spririva
Bullous emphysema	Wheezing	Advair
Pulmonary congestion	Smoking	Oxygen

Bronchitis	Chest pains	Albuterol
Acute respiratory failure	Cough	Combivent
Asthmas	Reflux	Prednisone
Gastro esophageal reflux	Crackle	Atrovent
Carcinoma of lung	Clubbing (morphologic abnormality)	Medrol
Pneumonia	Carbon dioxide, increased level	DuoNeb
Congestive Heart Failure	Deficiencies, Oxygen	Daliresp

Table 4.8 Selection of Ground-Truth Terms

Precision and recall were used at the primary performance metrics. Relevant terms are those defined in the ground truth dictionary and retrieved terms are those found by using both baseline and COED methods. The number of relevant terms is fixed for each category of medical terms. However, the number of retrieved terms is varied where $10 \leq n \leq relevent\ terms$ and $n \in \mathbb{Z}$. Precision and recall are defined in eq. (4.19) and eq. (4.20).

$$precision = \frac{|\{relevent\ terms\} \cap \{retrieved\ terms\}|}{|\{retrieved\ terms\}|} \quad \textbf{(4.19)}$$

$$recall = \frac{|\{relevent\ terms\} \cap \{retrieved\ terms\}|}{|\{relevent\ terms\}|} \quad \textbf{(4.20)}$$

CHAPTER 5: RESULTS

5.1 Readmission Prediction using Natural Language Processing

Tables 5.1-5.4 show a selection of the top features found for each feature selection algorithm.

Feature	Description
Stent	Medical device to enable free flow of blood vessels.
C0034072	Pulmonary Heart Disease
C0033036	Atrial Ectopic Beat
Valium	Anti-anxiety drug of the Benzodiazepine family.
CPAP	Medical device to assist in patient breathing.
Bronchoscopy	Diagnostic technique to analyze airways.

Table 5.1 Selection of Features Discovered by Forward Selection Wrapper

Feature	Description
C0018099	Gout
Cardiomyopathy	Abnormal heart muscle which may have difficulty pumping blood.
Risperdal	Drug to treat bipolar disorder.
Marginal	Descriptive term used in conjunction with diseases.
Hospice	End of life care facility.
C0728797	Flexeril is a muscle relaxant.

Table 5.2 Selection of Features Discovered by CFS

Feature	Description
Cranial	Descriptive term relating to the skull.
C0008679	Chronic disease. Used in conjunction with a specific disease.
C0037005	Shoulder dislocation.
C2710117	Drug used to treat low sodium levels in heart and liver disease patients.
Cardiomyopathy	Abnormal heart muscle which may have difficulty pumping blood.
Discharges	Flow of fluids from the body.

Table 5.3 Selection of Features Discovered by GR

Feature	Description
C0018099	Gout
Defibrillator	Medical device to correct ventricular fibrillation
Allopurinol	Drug to tree excess uric acid in the blood.
C0018802	Congestive Heart Failure
Respirdal	Drug to treat bipolar disorder.
Medtronic	Medical device manufacturer.

Table 5.4 Selection of Features Discovered by CS

Unlike GR and CS, wrapper and CFS methods choose an optimal subset of features. Table 5.5 shows that of the 5,428 features in the full dataset, wrapper chose an average of 292 features and CFS chose 125. A paired t-test was performed and a p value of < 0.01 was found. Thus, the two methods found a statistically significant different number of features to be optimal.

Fold #	Wrapper	CFS
1	274	134
2	281	129
3	343	140
4	239	123
5	309	134
6	172	131
7	376	124
8	359	129
9	340	68
10	232	140
Average	292	125

Table 5.5 Comparison of Optimal Number of Features Discovered by Wrapper and CFS

An analysis of intersecting features was performed and Table 5.6 shows the degree with which each feature selector overlaps, averaged over 10 folds. For CS, the optimal subset of features was found to be 795 and was used for comparison. For GR, the optimal subset of features was found to be 880. Feature selectors which discover the same features can be considered similar. Most algorithms share around 0.25 or less of the same features in the optimal set. However, CS and GR share most of the same features. Although these

methods are both statistical methods, they use different mathematical methods to arrive at their results.

Algorithm				
	Wrapper	**CFS**	**GR**	**CS**
Wrapper	X	0.256	0.154	0.153
CFS	X	X	0.153	0.153
GR	X	X	X	0.882
CS	X	X	X	X

Table 5.6 Overlap of Features Between Feature Selectors

5.1.1 AUC

An analysis of the AUC of classifiers was performed and presented in Table 5.7 AUC of Classifiers as Grouped by Feature Selector. The overall best classifier was RF, with NB a close second. The best feature selection method was CFS with CS and GR a close second and third. Surprisingly, wrapper using forward selection did not perform well. One possibility is the feature subset produced was overfit.

Feature Selector	**NB**	**RF**	**SVM**	**kNN**	**Average**
None	0.603	0.657	0.567	0.632	0.614
Wrapper	0.596	0.674	0.547	0.617	0.608
CFS	0.634	0.693	0.584	0.640	0.637
GR	0.688	0.646	0.547	0.635	0.629
CS	0.690	0.662	0.543	0.628	0.630
Average	0.6422	0.6664	0.5576	0.6304	
p-value	< 0.01	0.04	> 0.05	> 0.05	

Table 5.7 AUC of Classifiers as Grouped by Feature Selector

Great improvement of models over the baseline method (no feature selection) is seen. The feature selection method showing best improvement is analyzed using a paired t-test, with each fold representing an AUC value and a threshold of p=0.05 used for statistical significance. For NB, CS was tested against the baseline and CFS tested for all other algorithms. The best performing algorithms (RF and NB) have statistically significant improvement. SVM and kNN do not show statistically significant improvement using

feature selection models. However, those two algorithms performed poorly in comparison to RF and NB and would be discarded for the final framework regardless of statistical significance.

Finally, the statistical methods were iteratively compared using 10-fold cross validation and NB classifier while varying the number of features selected. The optimal number of features for CS and GR were around 795 and 880 (as shown in Figure 5.1), at which point the AUC began to decline. In comparison, as previously shown, wrapper and CFS methods found the optimal number of features to be 292 and 125.

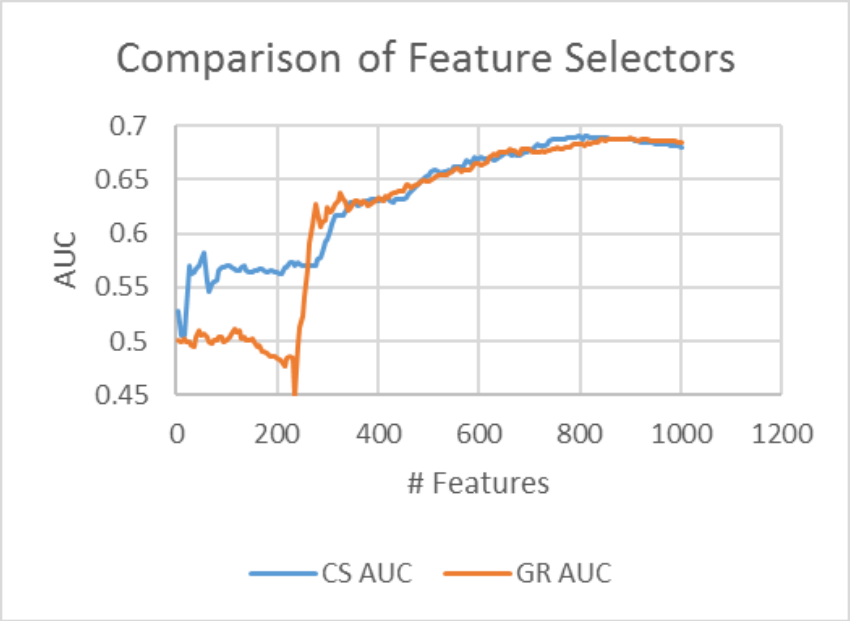

Figure 5.1 Effect Upon AUC Varying Number of Features for CS

5.1.2 Time

For many applications, model creation and instance classification time may be just as important as AUC and require a balance between speed and AUC. Table 5.8 shows model creation and test instance evaluation time, based on optimal number of features selected by the feature selector. NB is known to be an extremely fast classifier, and NB

with CS was previously shown to be the second highest AUC. This combination may be a good balance between speed and AUC performance. Additionally, CS chooses features extremely quickly. The highest AUC combination of RF with CFS shows model creation and classification time to be much longer. Additionally, Table 5.9 shows CFS to discover features much slower than CS. The small decrease in AUC may be acceptable for a readmission model which can choose features and build classifier in only a few seconds.

Feature Selector	NB	RF	SVM	kNN
None	8.48	8384.50	1120.70	313.47
Wrapper	5.75	6926.13	793.80	154.01
CFS	5.54	7469.47	842.38	160.41
GR	3.56	4063.11	632.32	75.17
CS	3.50	4488.26	604.24	70.53

Table 5.8 Model Creation and Evaluation Time (ms) Averaged Over 10 Folds

Feature Selector	NB
Wrapper	5.85 hours
CFS	8.11 minutes
GR	5.62 seconds
CS	4.90 seconds

Table 5.9 Feature Selection Algorithm Time Averaged Over 10 Folds

5.2 Cost

5.2.1 Per-instance cost

As shown in Table 5.10, the classifier with the best AUC is not necessarily the classifier which results in the lowest cost. RF obtained the highest AUC, but the lowest mean per-instance misclassification cost was obtained by NB by a large margin. Additionally, SVM and kNN obtained relatively low costs but poor AUC. Figure 5.2 shows a scatter plot of AUC in relation to cost. The Pearson correlation coefficient across all folds is -0.21 and shows there is little correlation between AUC and cost.

Classifier	AUC	Mean Instance Misclassification Cost
NB	0.643	$4,991.67

RF	0.657	$8,373.45
SVM	0.567	$6,540.98
kNN	0.574	$6,493.68
C4.5	0.546	$7,770.10
Bagging	0.626	$8,374.73
Boosting	0.574	$8,327.43

Table 5.10 Comparison of Per-Instance Cost and AUC

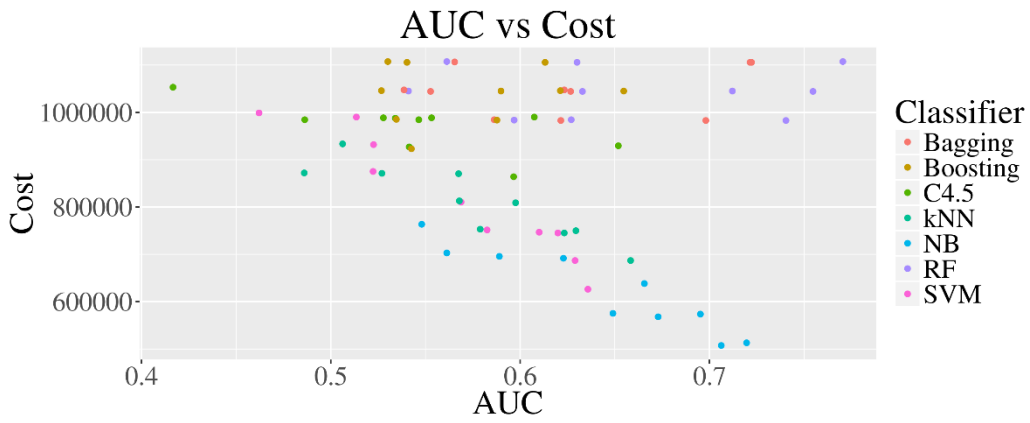

Figure 5.2 Scatter Plot Comparing CMS Penalty Cost and AUC

Cost-sensitive classification offers significant benefits as shown in Table 5.11. RF, Bagging, and Boosting greatly benefited from cost sensitive classification in comparison to cost insensitive classification.

Classifier	**Mean Instance Misclassification Cost**	
	Baseline	**Cost Sensitive Classification**
NB	$4,991.67	$4,189.65
RF	$8,373.45	$680.13
SVM	$6,540.98	$6,540.98
kNN	$6,493.68	$6,497.53
C4.5	$7,770.10	$1,833.61
Bagging	$8,374.73	$685.26
Boosting	$8,327.43	$685.26

Table 5.11 Comparison of Cost Sensitive Classification and Cost Insensitive Classification

5.2.2 Dataset cost

	Total Misclassification Cost

Classifier	Fixed λ	Updatable λ
NB	-$527,040.00	-$544,148.57
RF	-$32,994.29	-$862,445.71
SVM	-$300,685.71	-$300,685.71
kNN	-$307,588.57	-$307,108.57
C4.5	-$121,291.43	-$226,617.14
Bagging	-$32,834.29	-$701,017.14
Boosting	-$39,737.14	-$935,440.00

Table 5.12 Comparison of Fixed and Updatable FN Cost

When analyzed in batches, results show there is a significant benefit to updating cost parameters as new information becomes available (as shown in Table 5.12). Bagging is able to decrease cost 20x and most other classifiers able to make significant gains in misclassification cost.

5.2.3 Cost Reduction

Two baseline methodologies were chosen for comparison. The first baseline is the NB classifier which ignores cost and performs a traditional classification task. Most systems reviewed by Kansagara et al. use a similar method of classification which ignores cost and serves a reasonable baseline. The second baseline method assumes to intervene on all patients, or All-Intervention (AI). While this is uncommon in practice, a comparison is helpful for reference purposes as there exists little data on this methodology when reducing HRRP cost. 10-fold stratified cross validation is performed and serves as a sampling method for which to build models. Final cost calculated and presented using above formulas as well as fraction of patients selected. Cost is reported as total cost for each stratified fold analysis, not per patient. These costs are relative and meant to be compared with baseline methods, not to be taken as absolute costs of a typical hospital. Obtaining a low penalty cost by intervening in a small number of patients is desired as this allows limited resources to be used elsewhere. Success rate of intervention and other

starting assumptions are shown in Table 5.13 and Table 5.14 to illustrate the effect upon results using various common scenarios.

C	\$10,000,000
C_{np}	\$10,000 * N
ω	1
P	1,000
$\overline{c_t}$	\$800

Table 5.13 Initial Variable Assumptions for All Scenarios

Assumption	**R**	$\boldsymbol{\rho}$	$\boldsymbol{\hat{\rho}}$	$\boldsymbol{p_s}$
A	143	0.14	.143	.90
B	143	0.14	.143	.97
C	205	0.20	.205	.90
D	205	0.20	.205	.97

Table 5.14 Variable Values Used for Each Assumption Scenario.

Assumption	**MinCost**	**Classification**	**AI**
A	\$11,920	\$18,640	\$96,720
B	\$11,040	\$18,640	\$96,720
C	\$6,720	\$18,640	\$96,720
D	\$6,160	\$18,640	\$96,720

Table 5.15 Cost Results Averaged Over 10-Folds for Each Assumption Scenario

The AI baseline methodology is shown in Table 5.15 to have significantly larger costs than all other methods. Table 5.16 illustrates the cost savings of MinCost vs baseline methods. The average penalty for MinCost is 51.93% lower than classification and 90.07% lower than AI. Assumptions C and D are shown to have the greatest cost savings, suggesting that hospitals with high readmission rates may benefit most from MinCost.

Assumption	**Classification**	**AI**
A	-36.05%	-87.67%
B	-40.77%	-88.58%
C	-63.94%	-93.05%
D	-66.95%	-93.63%
Average	-51.93%	-90.07%

Table 5.16 Percentage Cost Difference for MinCost vs Baseline Methodologies

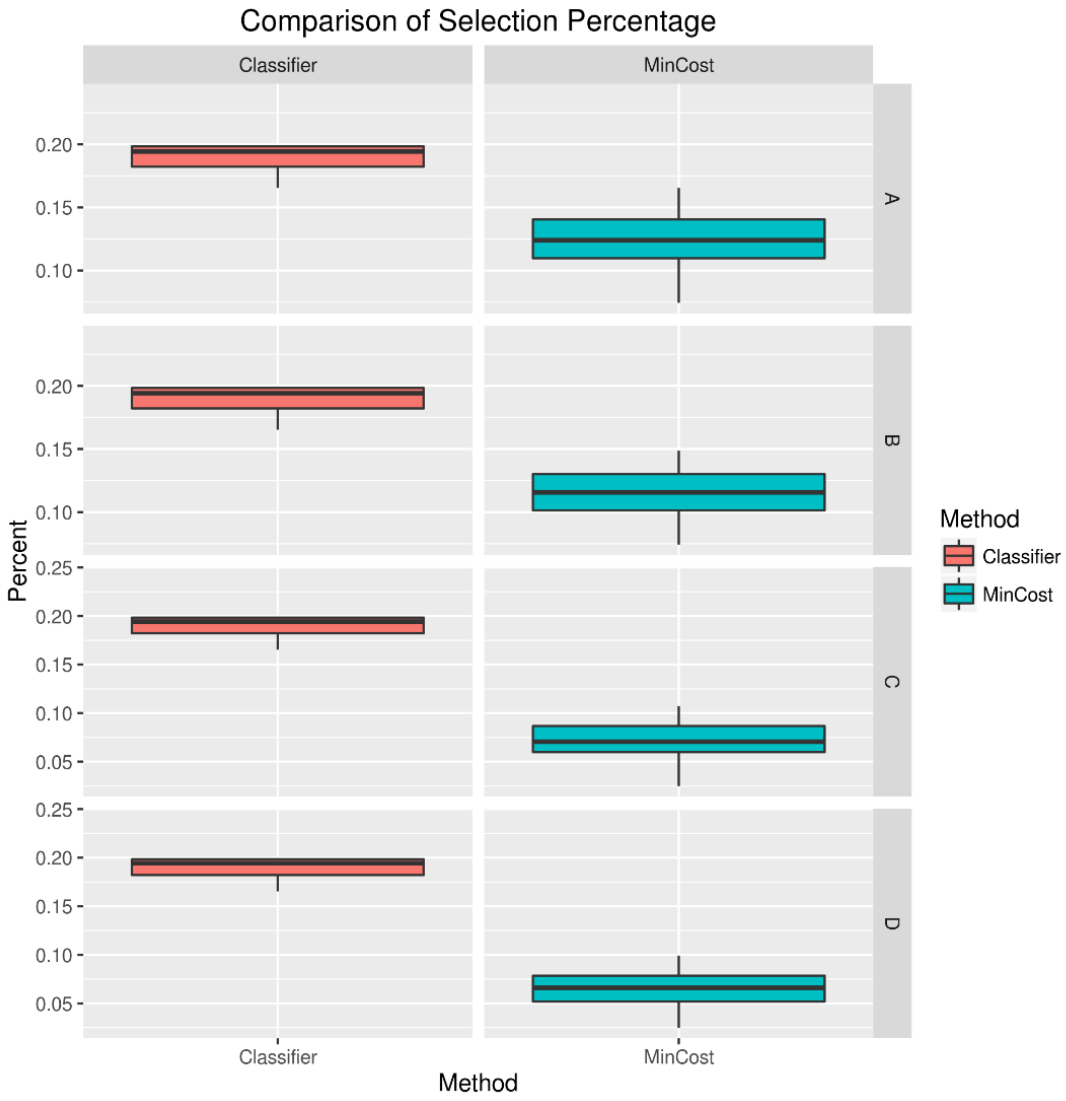

Figure 5.3 Comparison of MinCost and Binary Classification Patient Selection Percentage

As shown in Figure 5.3, baseline classification selects many more patients than necessary for readmission intervention. Average ERR for MinCost is -.001, however average ERR for baseline classification is -0.04. This is a significant increase. Many patients for baseline classification would have received a home healthcare professional while not actually lowering penalties. This would lead to an overall increase in costs either in monetary value or opportunity cost. Compared to binary NB classification, MinCost

significantly lowers net cost when all factors are taken into consideration (shown in Figure 5.4). These results are statistically significant using a paired t-test, where $p < 0.01$ in all instances.

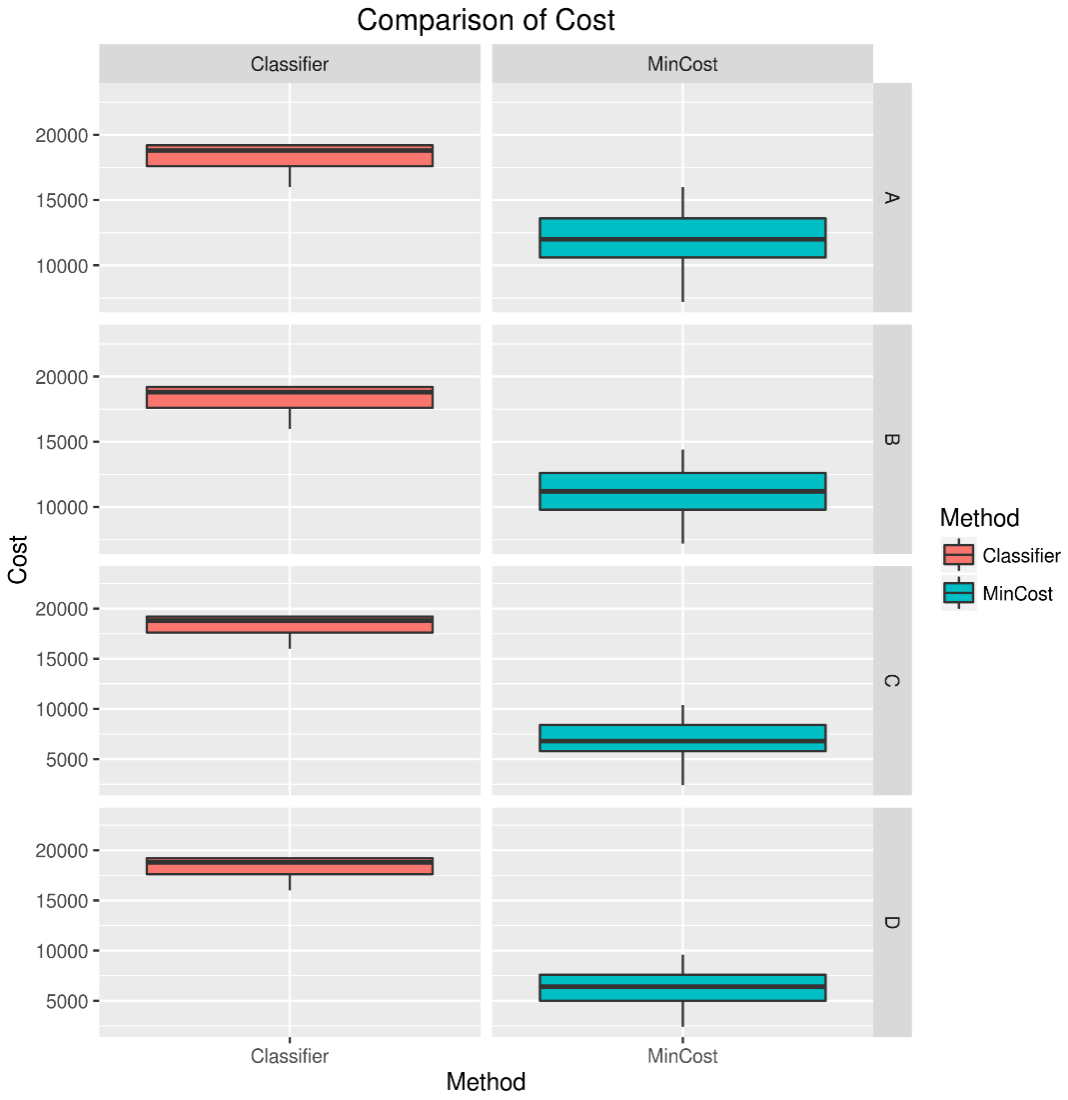

Figure 5.4 Comparison of MinCost and Binary Classification Patient Cost for Each Assumption Scenario

Success rate of patient intervention has shown to have an impact on cost as well. Figure 5.5 shows the cost impact of intervention success rate. A number of patients will need hospital readmission even after a home healthcare professional has visited. Increasing

the success rates of these professionals has an impact as well and should not be ignored. Scheduling short follow up visits can potentially increase success rates driving costs even lower. Stratified 5-fold cross validation is used to sample 5 test datasets and intervention success rate iteratively increased from .85 to 1.0. All 5 sampled datasets show decrease in cost. Additionally, increasing intervention cost in an attempt to improve success rates still shows lowered overall cost. Therefore, it may be worthwhile to consider multiple visits if it can potentially increase intervention success rate. In practice, a success rate of 1.0 would rarely be achievable.

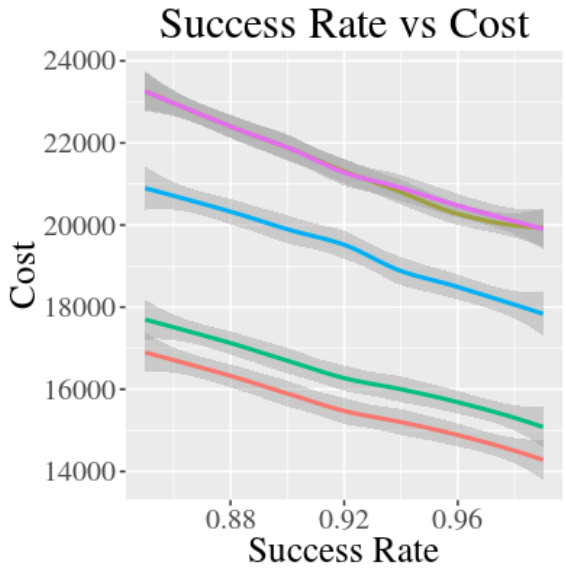

Figure 5.5 Comparison of Increasing Intervention Success Rate and Cost Under Assumption A

Many times reaching a zero ERR may not be possible due to a high initial ERR or small number of new patients under analysis. In those cases, Figure 5.6 shows NB classification to stop classifying patients for intervention far before optimal. When reaching a zero ERR is not possible, it may be most reasonable to send follow-up care to all or most of those patients in the DRG due to high costs of penalty. In this case, MinCost is reduced to the AI baseline. In practice, a medical facility may choose to initially only

intervene in extremely high risk patients while accumulating a pool of medium to high risk patients for calculation.

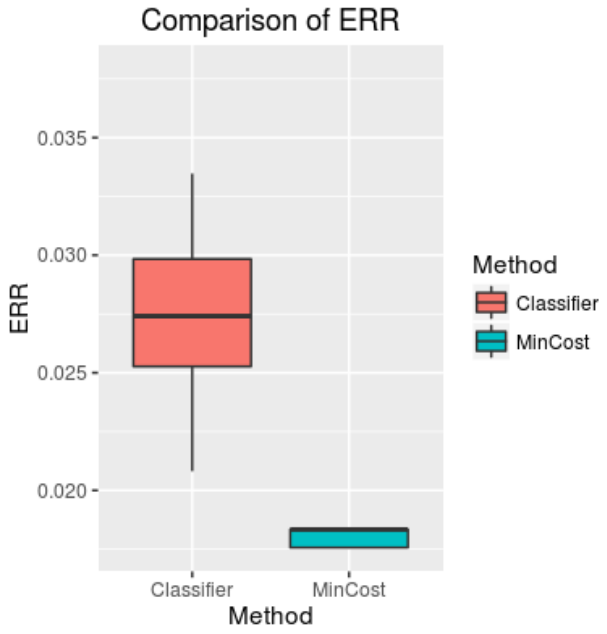

Figure 5.6 Comparison of ERR Under Cost Assumption A When too Few Patients Available to Reduce ERR to 0

5.3 Latest Topic Ensemble Learning

Table 5.17 shows that the base classifier with the lowest average cost using fixed cost methodology is LR. However, in practice NB may still be a good choice as it is considerably faster than LR. Additionally, the results were not statistically significant (p = 0.37) and therefore possible NB to be as good base classifier.

Classifier	Cost
NB	$796.36
kNN	$929.08
LR	$770.59
SVM	$5,885.19

Table 5.17 Comparison of Fixed Misclassification Cost Averaged Over 10 Folds for Each Base Classifier Using LTEL for Primary Hospital A

Figure 5.7 shows baseline methods plotted against LTEL. Each point represents a

hospital and cross-validation fold cost. NB was the only classifier tested due to time and computational resource limitations. NB builds classifiers relatively quickly while Table 5.17 and Table 5.18 provides evidence that NB is an acceptable choice. Hospitals containing more than 2,500 instances are plotted, however all available instances are used in building classifiers to maximize data usage. Hospitals containing fewer than 2,500 instances often did not contain enough positive instances to reliably perform 10-fold cross validation. A line starting at origin (0,0) with slope=1 overlays the plot. Points below the plot have a higher baseline cost when compared to LTEL and are considered to have performed better than the baseline methodology.

		Method				
		Primary	**Prm+Aux**	**TrAdaBoost**	**Bagging**	**LTEL**
Classifier	**NB**	-$7,493	-$7,284	-$5,997	-$7,320	-$7,998
	kNN	-$5,152	-$5,330	-$4,611	-$5,396	-$5,633
	LR	-$6,278	-$6,412	-$3,849	-$6,382	-$6,341
	SVM	-$3,187	-$3,165	-$4,612	-$3,165	-$3,156

Table 5.18 Comparison of Base Classifiers and Baseline Methodologies for Hospital A Using Updatable Cost

LTEL performed better than the baseline methods in most instances. In the few instances LTEL did not perform better than primary baseline, results are often close to the linear overlay, suggesting little performance degradation occurred. LTEL performed better than TrAdaBoost for all data points, suggesting that LTEL is more appropriate than a popular existing transfer learning method for this domain.

The updatable cost methodology showed similar gains as fixed cost. Compared to primary, LTEL often had lower cost. For the data points where cost was not lower, often times the point was very close to the linear overlay, suggesting performance degradation to be a relatively rare occurrence. All data points were located below the linear overlay for TrAdaBoost, as was the case with fixed cost classification. Table 5.18 shows a comparison

of classifiers and baseline methodologies for hospital A. When compared to all available methods and base classifiers, LTEL using NB has the lowest cost. These results were statistically significant ($p < 0.01$). NB has additional desirable qualities such as fast classification performance. A hospital which implements the LTEL system using NB can potentially significantly lower CMS penalties when compared to other known methods.

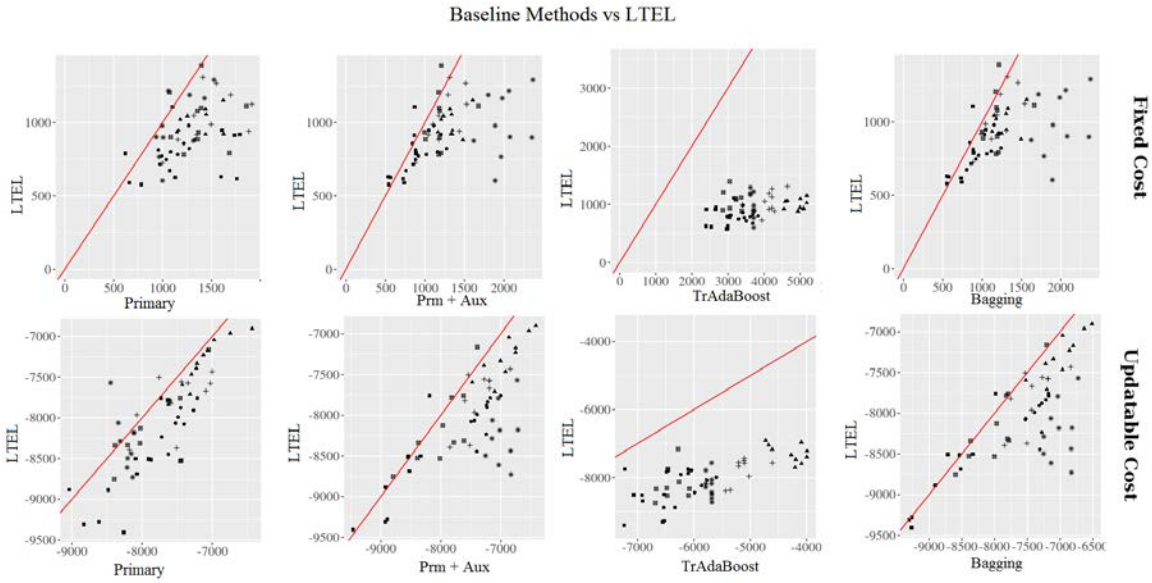

Figure 5.7 Scatterplots of LTEL Compared to Baseline Methods for Hospital A Using NB

5.4 Predicting Primary Cause of Readmission

Table 5.19 contains the top 25 features found using CS feature selection. These results agree with some static models such as LACE. DRG is found to be the most important predictor of readmission. This is an intuitive result as certain conditions such as heart disease and COPD are often associated with high rates of readmission. Additionally, many conditions may be associated with negative readmission such as minor injuries. Patients with many chronic conditions are also likely to need readmission as confirmed by the results.

Some features are highly correlated with negative readmission. Simple baby deliveries without complication do not require readmission in 98.92% of instances in this dataset. Elective readmissions are also predictive of negative readmission with 93.01% of elective admissions not requiring readmission. In contrast, 11.47% of non-elective admissions require readmission. Several specific conditions are also found to be predictive of readmission, such as Congestive Heart Failure (CHF) and end state renal disease. Finally, CS is shown to have a mix of DX[1-25] features and other non-bag-of-words features.

Feature	**Description**
DRG_NoPOA	DRG without present on admission flag
DRG	Diagnosis related group
MDC_NoPOA	MRD without present on admission flag
MDC	Major diagnosis group
NCHRONIC	Number of chronic conditions
SERVICELINE	Hospitalization type
NDX	Number of ICD-9-cm diagnoses
V270	Outcome of delivery, single liveborn
LOS	Length of stay
V5811	Encounter for antineoplastic chemotherapy
AGE	Age of the patient
DISPUNIFORM	Disposition of the patient at discharge
HCUP_ED	Patient required emergency services
PAY1	Indicator code of payer
9925	Heat exhaustion, unspecified
HOSP_NRD	Hospital identification code
ORPROC	Major operating room procedure was required
7359	Unspecified acquired deformity of toe
4280	Congestive heart failure
5856	End stage renal disease
7569	Unspecified anomalies of musculoskeletal system
ELECTIVE	Admission was elective
40391	Hypertensive chronic kidney disease
741	Spina bifida
V4511	Renal dialysis

Table 5.19 Highest Ranked Variables Discovered by CS

Table 5.20 contains the top 25 features found using GR feature selection. GR shows

a preference for features contained in ICD-9 diagnosis and procedure codes. Many features are related to pregnancy or delivery complications. Few features are found in both sets of results. Medical professionals may find features discovered by CS to be less opaque as many are intuitive and coincide with clinical research.

Feature	Description
V5811	Encounter for antineoplastic chemotherapy
9925	Heat exhaustion
64883	Abnormal glucose tolerance of mother
64403	Threatened premature labor
65103	Twin pregnancy
20400	Acute lymphoid leukemia
65423	Previous cesarean delivery, antepartum condition or complication
65963	Elderly multigravida, antepartum condition or complication
64913	Obesity complicating pregnancy
V270	Outcome of delivery, single liveborn
7359	Unspecified acquired deformity of toe
0392	Abdominal actinomycotic infection
64893	Conditions classifiable elsewhere of mother, antepartum condition or complication
7569	Unspecified anomalies of musculoskeletal system
2853	Antineoplastic chemotherapy induced anemia
V4512	Noncompliance with renal dialysis
64823	Anemia of mother, antepartum condition or complication
V652	Person feigning illness
66411	Second-degree perineal laceration
66401	First-degree perineal laceration
25063	Diabetes with neurological manifestations, type I
7309	Unspecified infection of bone
66331	Other and unspecified cord entanglement, complicating labor and delivery
64891	Other current conditions classifiable elsewhere of mother
64511	Post term pregnancy

Table 5.20 Highest Ranked Variables Discovered by GR

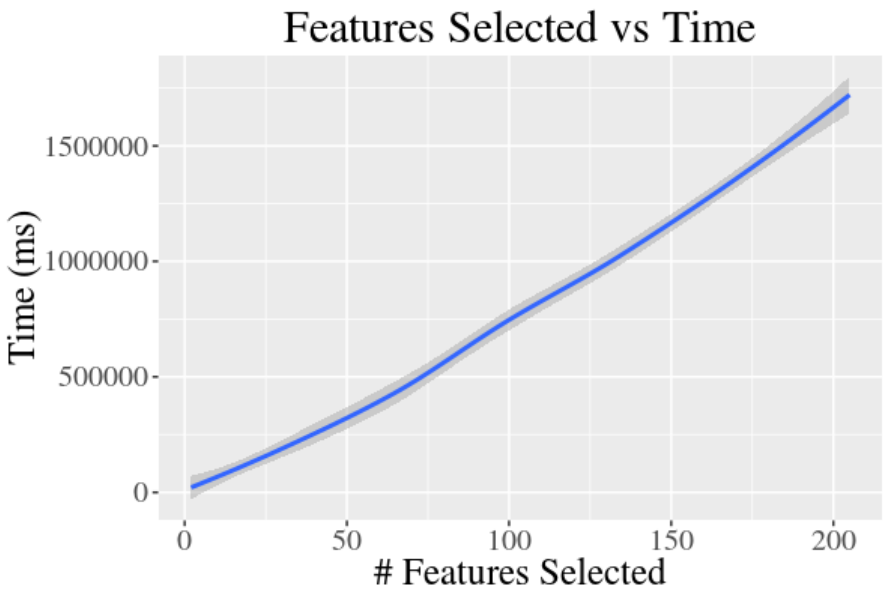

Figure 5.8 Plot of Model Creation Time vs Number of Features Selected

Figure 5.8 shows model creation time vs the number of features selected. NB appears to approximately scale linearly regarding the number of features selected. Figure 5.9 shows AUC performance vs the number of features selected. CS performs significantly better than GR. As shown in Table 5.21, these improvements are statistically significant. Peak performance for CS is around 50 features. Using a reduced subset of features may allow for considerably faster model creation time. For example, using 50 features vs 200 features would allow a model to be trained in approximately a quarter the time while maintaining similar AUC performance.

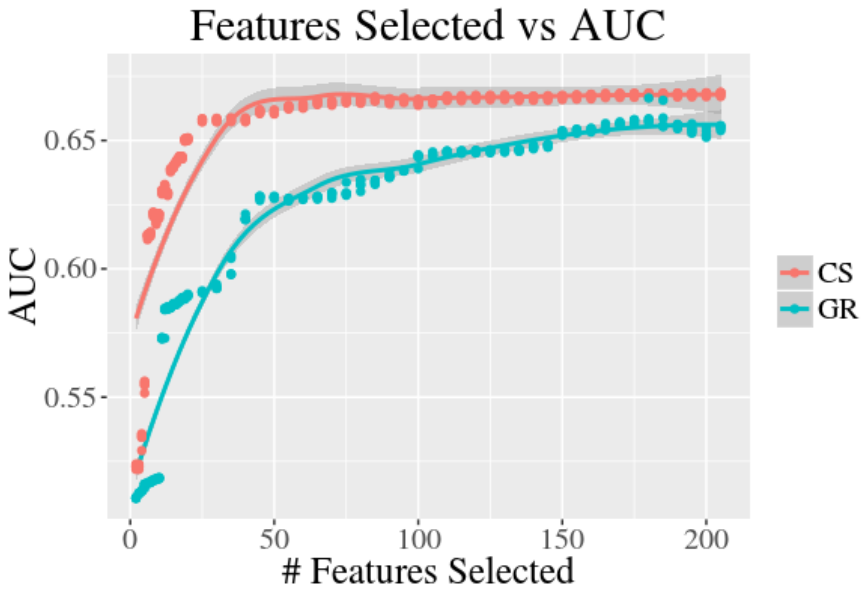

Figure 5.9. Scatterplot with Smoothed Average of AUC vs Number of Features Selected for NB Classifier

# Features	CS	GR	p-value
2	0.523004	0.510732	< 0.01
3	0.522989	0.512495	< 0.01
4	0.533817	0.513487	< 0.01
5	0.554641	0.515751	< 0.01
10	0.620839	0.518439	< 0.01
25	0.657886	0.591052	< 0.01
50	0.661141	0.627881	< 0.01
100	0.66486	0.64295	< 0.01
150	0.666922	0.653162	< 0.01
200	0.668016	0.65283	< 0.01

Table 5.21 AUC for Various Number of Selected Features

Finally, misclassification cost is shown in Figure 5.10. CS is shown to have lower per-instance misclassification cost than GR. Unlike AUC, the number of features found to be optimal settles around 175 features. Table 5.22 shows that for large numbers of features, these results are statistically significant.

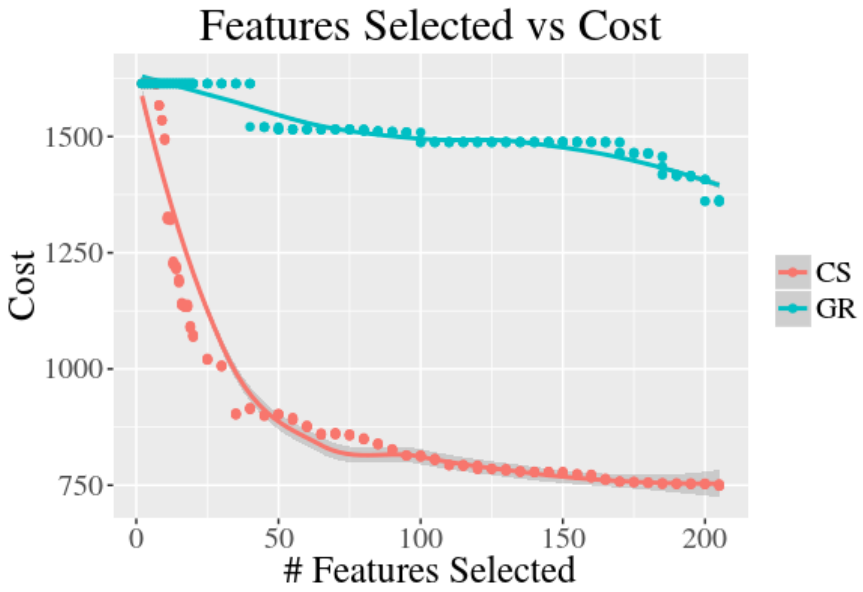

Figure 5.10. Scatterplot with Smoothed Average of Cost vs Number of Features Selected for NB Classifier

# Features	CS	GR	p-value
2	$1,614.04	$1,614.04	1
3	$1,614.04	$1,614.04	1
4	$1,614.04	$1,614.04	1
5	$1,614.04	$1,614.04	1
10	$1,494.23	$1,614.04	< 0.01
25	$1,020.70	$1,614.04	< 0.01
50	$902.50	$1,516.40	< 0.01
100	$812.86	$1,492.30	< 0.01
150	$776.41	$1,488.53	< 0.01
200	$752.67	$1,389.01	< 0.01

Table 5.22 Misclassification Cost for Various Number of Selected Features

Next, the LTEL algorithm is compared to baseline methods. NB as a standalone classifier, as well as bagging ensemble methods with NB as a base classifier are compared to LTEL. Table 5.23 shows the results using AUC as the primary performance metric. LTEL is able to increase the AUC a statistically significant amount. 5 LDA topics were extracted using LTEL and 5 bags were created for bagging. Bagging often works well with unstable learners and NB standalone performing the same as bagging is consistent with

that observation. Table 5.24 shows LTEL is able to decrease average misclassification cost a statistically significant amount as well.

Fold	NB	Bagging	LTEL
1	0.6662	0.6662	0.7024
2	0.6674	0.6674	0.6950
3	0.6676	0.6676	0.7006
4	0.6675	0.6675	0.7014
5	0.6664	0.6664	0.6938
Mean	0.6670	0.6670	0.6986
S.D.	0.0005	0.0005	0.0035

Table 5.23 Comparison of AUC for LTEL to Baseline Methods

Fold	NB	Bagging	LTEL
1	$1,222.95	$1,224.10	$851.15
2	$1,215.39	$1,216.26	$896.82
3	$1,218.87	$1,220.83	$941.23
4	$1,220.06	$1,221.17	$926.71
5	$1,226.61	$1,228.33	$946.61
Mean	$1,220.776	$1,222.13	$912.50
S.D.	$3.79	$3.98	$35.21

Table 5.24 Comparison of Cost for LTEL to Baseline Methods

Finally, primary cause of readmission is analyzed (shown in Table 5.25). AUC is used as the primary performance metric. LTEL is used at the classification mechanism. Prediction of readmission cause is only attempted for instances classified as readmission. MDCs with a high rate of same-cause readmission are classified correctly most often. Simple pregnancies rarely require readmission. When patients require readmission, it is often related to the pregnancy. However, MDCs with a medium amount of same-cause readmission (approximately 54% each) such as drug abuse and HIV infections are also often correctly classified. MDCs often associated with chronic diseases such as respiratory and kidney disease have lower, but still acceptable performance. Finally, infectious disease has only a slight class imbalance (27.8% same-cause readmission) but considerably lower

performance.

MDC	Description	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	S . D .
0	Pre-MDC	0.741	0.731	0.764	0.761	0.761	0.752	0.013
1	Nervous System	0.697	0.689	0.689	0.691	0.691	0.691	0.003
2	Eye	0.671	0.586	0.64	0.614	0.647	0.632	0.029
3	Ear, Nose, Mouth And Throat	0.599	0.601	0.609	0.597	0.61	0.603	0.005
4	Respiratory System	0.739	0.727	0.725	0.733	0.736	0.732	0.005
5	Circulatory System	0.777	0.77	0.769	0.773	0.777	0.773	0.003
6	Digestive System	0.673	0.654	0.654	0.656	0.673	0.662	0.009
7	Hepatobiliary System And Pancreas	0.831	0.815	0.819	0.827	0.825	0.823	0.006
8	Musculoskeletal System And Connective Tissue	0.672	0.647	0.648	0.657	0.659	0.657	0.009
9	Skin, Subcutaneous Tissue And Breast	0.687	0.685	0.678	0.672	0.686	0.682	0.006
10	Endocrine, Nutritional And Metabolic System	0.73	0.712	0.709	0.725	0.723	0.720	0.008
11	Kidney And Urinary Tract	0.733	0.721	0.721	0.725	0.731	0.726	0.005
12	Male Reproductive System	0.846	0.832	0.826	0.821	0.839	0.833	0.009
13	Female Reproductive System	0.841	0.844	0.818	0.844	0.835	0.836	0.010
14	Pregnancy, Childbirth And Puerperium	0.951	0.967	0.963	0.958	0.95	0.958	0.007
15	Newborn And Other Neonates (Perinatal Period)	0.979	0.988	0.989	0.98	0.978	0.983	0.005
16	Blood and Blood Forming Organs and Immunological Disorders	0.741	0.746	0.728	0.744	0.746	0.741	0.007
17	Myeloproliferative DDs (Poorly Differentiated Neoplasms)	0.922	0.924	0.927	0.934	0.922	0.926	0.004
18	Infectious and Parasitic DDs (Systemic or unspecified sites)	0.668	0.655	0.654	0.656	0.66	0.659	0.005
19	Mental Diseases and Disorders	0.896	0.883	0.884	0.884	0.897	0.889	0.006
20	Alcohol/Drug Use or Induced Mental Disorders	0.919	0.923	0.908	0.911	0.922	0.917	0.006
21	Injuries, Poison And Toxic Effect of Drugs	0.609	0.596	0.614	0.607	0.612	0.608	0.006
22	Burns	0.876	0.708	0.911	0.732	0.784	0.802	0.079
2	Factors Influencing Health	0.598	0.585	0.598	0.595	0.623	0.600	0.013

3	Status and Other Contacts with Health Services							
24	Multiple Significant Trauma	0.671	0.616	0.617	0.698	0.594	0.639	0.039
25	Human Immunodeficiency Virus Infection	0.947	0.896	0.92	0.945	0.936	0.929	0.019

Table 5.25 AUC of predicted readmission MDC codes.

5.5 Co-Occurring Evidence Discovery

5.5.1 Diseases & Disorders

A sample of the highest scoring diseases is shown in Table 5.26. The baseline method shows diseases which have a high population prevalence (such as diabetes), to occur higher in the baseline method than COED. Additionally, respiratory failure is a more appropriate highest rank term than hypertension. Diseases with a high population prevalence may still rank high in COED. For example, diabetes ranks as the fifth highest term. However, the goal of COED is not to completely eliminate frequently occurring diseases from retrieval results, only rank them lower by penalizing their prevalence in the general population. Precision and recall are higher in comparison to the baseline method as shown in Figure 5.11 and Figure 5.12.

Rank	**Baseline**	**COED**
1	Hypertension	Respiratory failure
2	Diabetes mellitus	Hypertension
3	Coronary disease	Pneumonia
4	Heart fibrillation	Congestive heart failure
5	Arteriopathic disease	Diabetes mellitus
6	Congestive heart failure	Chronic respiratory failure
7	Pneumonia	Acute respiratory distress
8	Respiratory failure	Acute chronic respiratory failure
9	Anemia	Chronic respiratory insufficiency
10	Kidney disease	Heart Fibrillation

Table 5.26 Selection of Top 10 Results for Diseases & Disorders

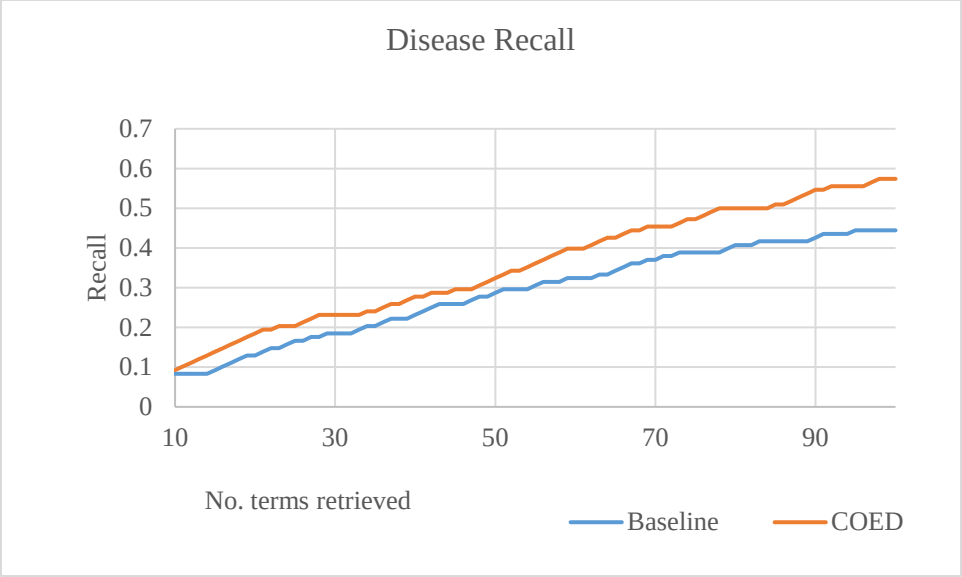

Figure 5.11 Comparison of Disease Recall for Baseline and COED Methodologies

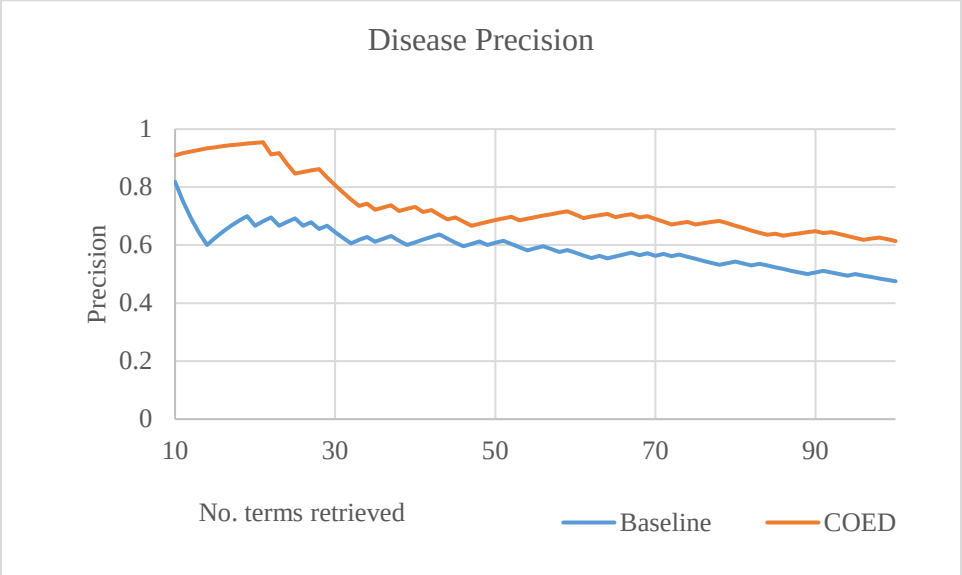

Figure 5.12 Comparison of Disease Precision for Baseline and COED Methodologies

5.5.2 Symptoms

A sample of the highest scoring symptoms is shown in Table 5.27. The baseline method returns the top scoring term as pain while COED returns a breathing condition. Additionally, COED returns smoking, a direct known cause of COPD, in the top results. Allergies are very common and appear in the baseline methodology but do not appear in

the selection of COED results. Precision and recall additionally are higher in comparison to the baseline method as shown in Figure 5.13 and Figure 5.14.

Rank	Baseline	COED
1	Pain NOS	Dyspneas
2	Dyspneas	Oxygen supply
3	MG body	Wheezings
4	Normal skin	Pain NOS
5	Chest pains	MG body
6	Cough	Respiratory insufficiency
7	Allergies	Smoker
8	Arterial tension	Decreased air entry
9	Edema	Cough
10	Atrial fibrillations	Normal skin

Table 5.27 Selection of Top 10 Results for Symptoms

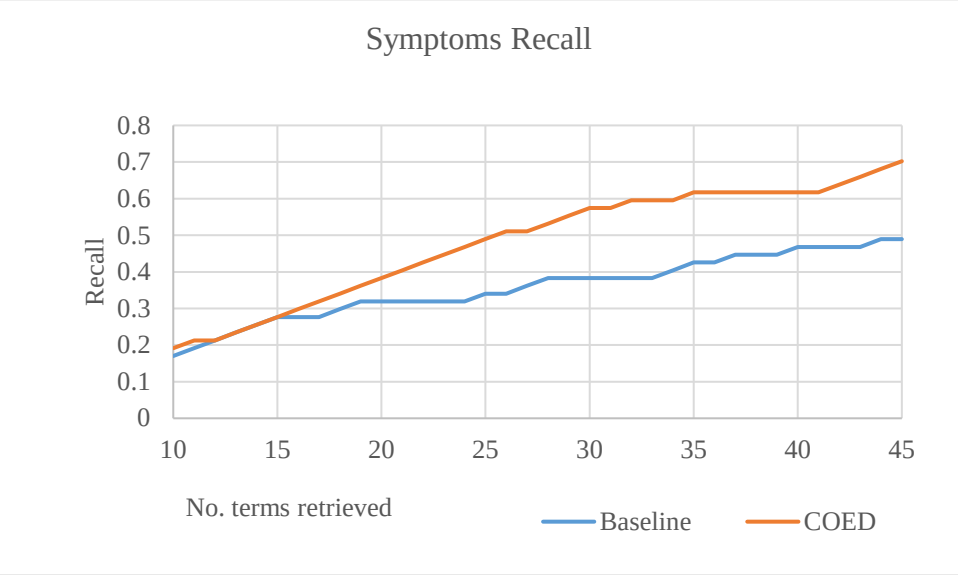

Figure 5.13 Comparison of Symptoms Recall for Baseline and COED Methodologies

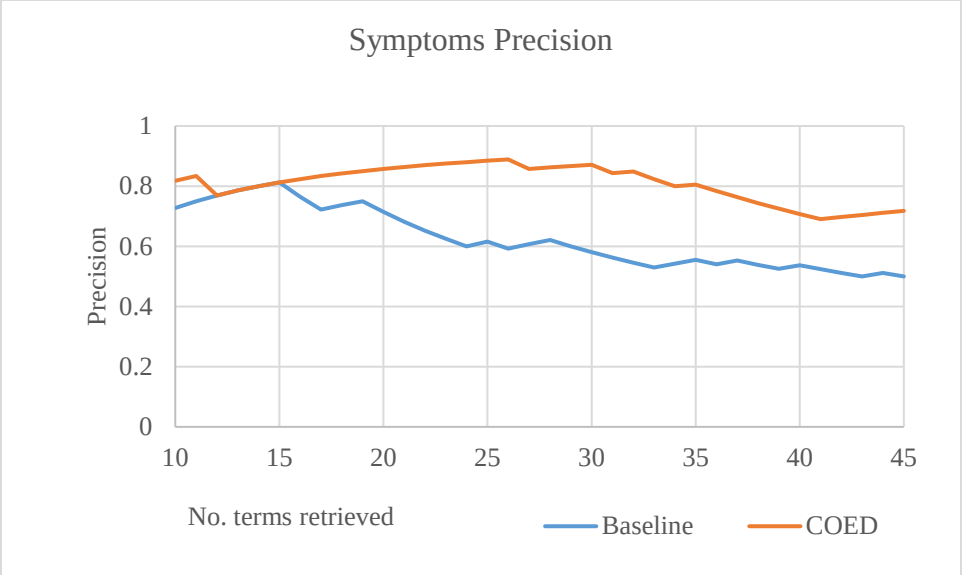

Figure 5.14 Comparison of Symptoms Precision for Baseline and COED Methodologies

5.5.3 Medications

A sample of the highest scoring medications is shown in Table 5.28. The baseline method has chosen Aspirin as the highest scoring medication. However, Aspirin is a very common medication and population prevalence causes it to be highly ranked. COED has chosen Spriva, a popular medicine to treat bronchospasms caused by COPD. In contrast to diseases and symptoms, COED and baseline methods are much more similar and precision and recall much closer in value. Precision and recall additionally are higher in comparison to the baseline method as shown in Figure 5.15 and Figure 5.16.

Rank	**Baseline**	**COED**
1	Aspirin	Spiriva
2	Albuterol	Advair
3	Oxygen	Oxygen
4	Advair	Albuterol
5	Prednisone	Combivent
6	Marevan	Prednisone
7	Lisinopril	Atrovent
8	Medrol	Medrol
9	Combivent	DuoNeb

10	Spiriva	Daliresp

Table 5.28 Selection of Top 10 Results for Medications

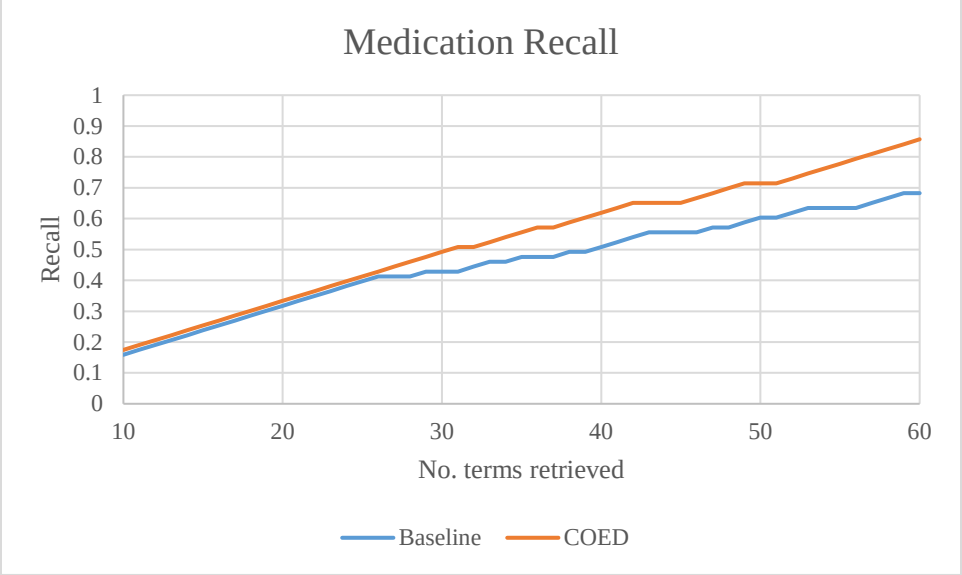

Figure 5.15 Comparison of Medication Recall for Baseline and COED Methodologies

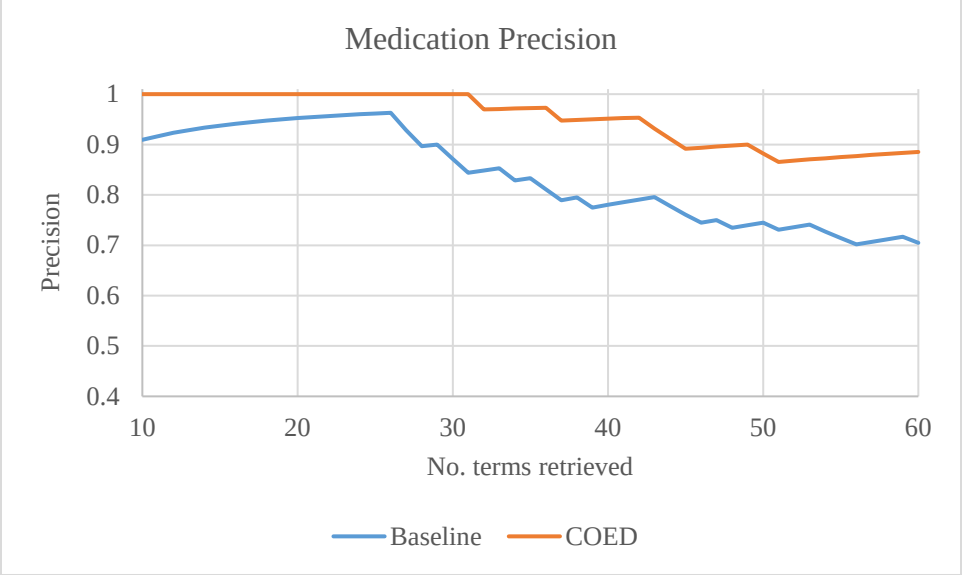

Figure 5.16 Comparison of Medication Precision for Baseline and COED Methodologies

CHAPTER 6: CONCLUSIONS

6.1 Summary

Predictive analytics of hospital readmission is an important research area with many open questions and potential for improvement. This research addressed many of these open questions and presented several research methodologies with positive results.

6.2 Readmission Prediction using Natural Language Processing

Our readmission analysis system represents a natural language approach to patient readmission prediction. Components were evaluated and it was found that using NB classifier with CS, selecting around 15% of the full feature set to be most effective. The system was able to predict hospital readmissions using only bag-of-words representation and UMLS annotations at least as well as current structured systems and in many cases, better than existing systems. Our approach offers the advantage that separate data collection is not required for readmission prediction since clinical notes are already collected by medical institutions. Additionally, unstructured data requires no data format conversions to be evaluated by an external system. Structured systems using RDBMS typically require many data conversion steps to reach an expected data format. Thus, our system presents easy integration into existing EHR systems.

With the increase in EHR systems, clinical notes will become increasingly important and NLP techniques will need to be considered when creating decision support systems. The results have shown the importance of feature selection and model creation time to the implementation of practical systems.

6.3 Cost

Current systems have generally disregarded cost when creating and evaluating PHRS. However, hospital administrators may prefer cost metrics when classifying and evaluating potential hospital readmissions, due to limited resource availability. Cost sensitive classification methods which directly incorporate CMS penalties was presented. This work was unique due to the rate-based nature of these penalties which are rare in many domains. AUC and cost were shown to have little correlation, suggesting AUC may not be the most appropriate performance metric for this domain. This result is important as most current systems use AUC as the primary performance metric. Two approaches to cost classification were presented and although nightly batch processing is shown to have considerably lower costs, evaluation at discharge may offer practical benefits which hospitals prefer. As the patient is still in immediate contact, providing additional care may be considerably simpler. Both methods were shown to have performance gains over cost-insensitive classification.

Additionally, these costs may be considerably reduced by finding the optimal set of patients to intervene. Rather than using binary classification, leveraging the probability of readmission was shown to have significant benefit. During nightly batch processing, sorting patients by probability of readmission allows the patients most likely to be readmitted to be addressed first. When ERR reaches zero or cost of intervention is more than cost of non-intervention, additional patients are not selected. This method allows for a more intelligent allocation of resources and was shown to have significant cost savings.

6.4 Latent Topic Ensemble Learning

Little research has been done in combining data from many medical facilities.

Simply combining data from many institutions may result in a poor model. Using topic modeling a methodology was presented which was shown to have cost savings when compared to several baseline methodologies. These baseline methodologies included using data from a target hospital, combining all available data, and a popular transfer learning algorithm known as TrAdaBoost. This research potentially represents the first methodology which is able to intelligently combine data from many hospitals.

To overcome this limitation, we proposed a latent topic based ensemble learning framework. We argued that when building a hospital re-admission prediction model, the data distributions across different hospitals vary significantly. Existing methods often combine all data together, or select a small subset of samples from all available data, which result in biased or ineffective models. Alternatively, we proposed LTEL to derive latent topics from different hospitals, and use topics to align data across hospitals and determine weight accordingly. The weighted instances from different hospitals are then combined to build classifier to predict instances from a primary hospital. The experiments and validations from data collected from 16 regional hospitals demonstrated significant cost reduction compared to best performing baseline available.

6.5 Predicting Primary Cause of Readmission

Predicting the primary cause of readmission has previously received little published research. However, many patients are readmitted for a different reason than index admission and knowing this information may be invaluable to clinical staff. Due to lack of readmission diagnostic codes in the unstructured dataset, a structured dataset was chosen. The HCUP NRD dataset provided two primary research opportunities. First, the dataset has seen little research in using the full dataset for readmission research. A set of proposed

baseline methodologies was outlined to utilize this large dataset and it was shown CS feature selection to be appropriate while selecting approximately 175 features to retain predictive cost quality. When using AUC as the primary metric, CS using 50 features was most appropriate. Second, predicting the primary cause of readmission was presented. Results showed that some MDCs with high same-readmission causes (those related to pregnancy) to have the greatest AUC and most easily classified. However, other MDCs with a medium same-readmission cause (such as HIV) were still able to predict with high discriminative ability. MDCs with extremely low same-readmission cause performed poorly. The two categories of readmission for which federal readmission legislation exists is chronic disease and infectious disease. Chronic disease primary cause readmission showed reasonably good discriminative ability. However, infectious disease prediction is more difficult and requires further research into readmission models. Overall, primary cause of readmission prediction showed good results which may be useful to clinical staff.

Future research may utilize additional algorithms in a big data environment. NB was chosen throughout this research due to its computationally fast performance. Previous research showed NB to be an appropriate algorithm choice for the unstructured dataset, but others, such as RF may provide improvements.

6.6 Co-Occurring Evidence Discovery

Finally, an improved methodology for creating disease dictionaries was presented. Currently, indexes such as the Charlson co-morbidity index are used as a feature for readmission classification. Many times these indexes are built using simple co-occurrence definitions. Additionally, as part of a CDSS medical staff may wish to find diseases, medications, and symptoms associated with a specific disease. COPD was chosen as a

baseline disease and ground-truth dictionary assembled. Results showed improvements over existing methods. Additionally, big data approaches are possible in finding evidence co-occurrence and COED implemented in a big data environment using Apache Spark.

As shown in the results, penalizing terms which are highly frequent in the corpus results in better precision and recall performance. Penalizing frequently occurring terms gives a better picture of the diseases, symptoms, and medications co-occurring with COPD. Using a mathematical and computational approach rather than purely expert driven approach, large dictionaries of COPD related terms can be assembled in a short amount of time. Additionally, localized data may return slightly different results based on patient population. This allows dictionaries to be created on a per-hospital basis rather than nationally, which may not account for localized concerns.

Future work intends to expand this methodology to other diseases to increase confidence in results. Many diseases do not contain ground truth dictionaries for the purposes of information retrieval analysis and must be created using similar methodology. Finally, we intend to integrate the software into an EHR system directly for analytical feedback to medical professionals about their patient population. This can serve as a decision support system to assist medical staff in developing patient treatment procedures.

6.7 Future Work

The presented systems show a step forward in PHRS research. However, additional work is possible. Due to limitations in data availability, merging of structured and unstructured was not possible. Merging the two data sources and performing feature selection under cost-sensitive classification and evaluation may yield improvements. A framework using progressive enhancement may allow structured data to be integrated when

available, defaulting to purely unstructured data.

Additionally, implementation of the cost-sensitive approaches in partnership with a large payer such as Medicare would be desirable. Measurement of long term accuracy of a cost-sensitive system would be required to enhance confidence in the previously acquired results. Replacing currently available readmission scoring systems such as readmission.org with our methodology available as web services may enable hospitals to quickly integrate a high quality PHRS into current operations.

Finally, integration of COED into an existing CDSS may yield additional improvements. COED offers the attending physician important information regarding co-occurring diseases, symptoms, and medications. Implementation of COED in a hospital environment may allow for feedback regarding which treatments work best for lowering readmission. Additionally, the construction of a new co-morbidity index using COED which integrates diseases, medications, and symptoms should be explored. Although such a system would no longer be just a co-morbidity index, the index may offer additional information and potentially improved results over existing co-morbidity indexes.

BIBLIOGRAPHY

[1] R. N. Axon and M. V Williams, “Hospital readmission as an accountability measure,” *Jama*, vol. 305, no. 5, pp. 504–505, 2011.

[2] Centers for Medicare and Medicaid Services, “National Health Expenditure Accounts,” 2016. [Online]. Available: http://go.cms.gov/1UFHHer. [Accessed: 25-Jan-2017].

[3] B. R. Furrow, “An overview and analysis of the impact of the emergency medical treatment and active labor act,” *J. Leg. Med.*, vol. 16, no. 3, pp. 325–355, 1995.

[4] D. Goodman, E. Fisher, and C. Chang, “The Revolving Door: A Report on US Hospital Readmissions,” *Princeton, NJ Robert Wood Johnson Found.*, 2013.

[5] Centers for Medicare and Medicaid Services, “Readmissions Reduction Program,” 2014. [Online]. Available: http://go.cms.gov/1gLbnoa. [Accessed: 15-Jun-2015].

[6] J. Hoffman, “Overview of CMS Readmissions Penalties for 2016,” 2015. [Online]. Available: http://www.besler.com/2016-readmissions-penalties/. [Accessed: 25-Sep-2016].

[7] C. Boccuti and G. Casillas, “Aiming for Fewer Hospital U-turns,” 2016. [Online]. Available: http://kaiserf.am/1KWsOkd. [Accessed: 25-Jan-2017].

[8] G. F. Anderson and E. P. Steinberg, “Hospital readmissions in the Medicare population,” *N. Engl. J. Med.*, vol. 311, no. 21, pp. 1349–1353, 1984.

[9] G. F. Anderson and E. P. Steinberg, “Predicting hospital readmissions in the Medicare population,” *Inquiry*, vol. 311, no. 21, pp. 251–258, 1985.

[10] R. S. Phillips, C. Safran, P. D. Cleary, and T. L. Delbanco, “Predicting emergency readmissions for patients discharged from the medical service of a teaching hospital,” *J. Gen. Intern. Med.*, vol. 2, no. 6, pp. 400–405, 1987.

[11] E. F. Philbin and T. G. DiSalvo, “Prediction of hospital readmission for heart failure: development of a simple risk score based on administrative data,” *J. Am. Coll. Cardiol.*, vol. 33, no. 6, pp. 1560–1566, 1999.

[12] P. E. Cotter, V. K. Bhalla, S. J. Wallis, and R. W. S. Biram, “Predicting readmissions: Poor performance of the LACE index in an older UK population,” *Age Ageing*, vol. 41, no. 6, pp. 784–789, 2012.

[13] H. Wang, R. D. Robinson, C. Johnson, N. R. Zenarosa, R. D. Jayswal, J. Keithley, and K. A. Delaney, “Using the LACE index to predict hospital readmissions in congestive heart failure patients,” *BMC Cardiovasc. Disord.*, vol. 14, no. 1, pp. 1–8, 2014.

[14] S. Yu, F. Farooq, A. van Esbroeck, G. Fung, V. Anand, and B. Krishnapuram, “Predicting readmission risk with institution-specific prediction models,” *Artif. Intell. Med.*, vol. 65, no. 2, pp. 89–96, 2015.

[15] A. G. Au, F. A. McAlister, J. A. Bakal, J. Ezekowitz, P. Kaul, and C. Van Walraven, “Predicting the risk of unplanned readmission or death within 30 days of discharge after a heart failure hospitalization,” *Am. Heart J.*, vol. 164, no. 3, pp. 365–372, 2012.

[16] S. Sushmita, G. Khulbe, A. Hasan, S. Newman, P. Ravindra, S. B. Roy, M. De Cock, and A. Teredesai, “Predicting 30-Day Risk and Cost of ‘ All-Cause ’ Hospital Readmissions,” *Expand. Boundaries Heal. Informatics Using AI*, pp. 453–461, 2015.

[17] S. L. Hummel, P. Katrapati, B. W. Gillespie, A. C. DeFranco, and T. M. Koelling, “Impact of prior admissions on 30-day readmissions in medicare heart failure inpatients,”

Mayo Clin. Proc., vol. 89, no. 5, pp. 623–630, 2014.

[18] D. Kansagara, H. Englander, A. Salanitro, D. Kagen, C. Theobald, M. Freeman, and S. Kripalani, "CLINICIAN ' S CORNER Risk Prediction Models for Hospital Readmission A Systematic Review," *Jama*, vol. 306, no. 15, pp. 1688–1698, 2011.

[19] J. W. Thomas, "Does risk-adjusted readmission rate provide valid information on hospital quality?," *Inquiry*, pp. 258–270, 1996.

[20] E. A. Coleman, S. J. Min, A. Chomiak, and A. M. Kramer, "Posthospital care transitions: Patterns, complications, and risk identification," *Health Serv. Res.*, vol. 39, no. 5, pp. 1449–1465, 2004.

[21] A. Bottle, P. Aylin, and A. Majeed, "Identifying patients at high risk of emergency hospital admissions: a logistic regression analysis.," *J. R. Soc. Med.*, vol. 99, no. 8, pp. 406–14, 2006.

[22] E. F. R. Morrissey, J. C. McElnay, M. Scott, and B. J. McConnell, "Influence of drugs, demographics and medical history on hospital readmission of elderly patients," *Clin. Drug Investig.*, vol. 23, no. 2, pp. 119–128, 2003.

[23] K. Zolfaghar, J. Agarwal, D. Sistla, S.-C. Chin, S. Basu Roy, and N. Verbiest, "Risk-O-Meter: An Intelligent Clinical Risk Calculator," *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 1518–1521, 2013.

[24] A. Hosseinzadeh, M. Izadi, A. Verma, D. Precup, and D. Buckeridge, "Assessing the Predictability of Hospital Readmission Using Machine Learning," *Proc. Twenty-Fifth Innov. Appl. Artif. Intell. Conf. Assess.*, pp. 1532–1538, 2013.

[25] P. Braga, F. Portela, M. F. Santos, and F. Rua, "Data mining models to predict patient's readmission in intensive care units," *6th Int. Conf. Agents Artif. Intell. ICAART*

2014, vol. 1, no. Im, pp. 604–610, 2014.

[26] R. Duggal, S. Shukla, S. Chandra, B. Shukla, and S. K. Khatri, "Predictive risk modelling for early hospital readmission of patients with diabetes in India," *Int. J. Diabetes Dev. Ctries.*, 2016.

[27] D. D. Lewis, "Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval," *Machine Learning: ECML-98*. pp. 4--15, 1998.

[28] I. Rish, J. Hellerstein, and T. Jayram, "An analysis of data characteristics that affect naive Bayes performance," *Tec. Rep. RC21993, IBM Watson …*, 2001.

[29] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 5, pp. 1–35, 1999.

[30] I. Shams, S. Ajorlou, and K. Yang, "A predictive analytics approach to reducing 30-day avoidable readmissions among patients with heart failure, acute myocardial infarction, pneumonia, or COPD," *Health Care Manag. Sci.*, vol. 18, no. 1, pp. 19–34, 2015.

[31] J. Futoma, J. Morris, and J. Lucas, "A comparison of models for predicting early hospital readmissions," *J. Biomed. Inform.*, vol. 56, pp. 229–238, 2015.

[32] A. Agarwal, C. Baechle, R. Behara, and X. Zhu, "A Natural Language Processing Framework for Assessing Hospital Readmissions for Patients with COPD.," *IEEE J. Biomed. Heal. informatics*, 2017.

[33] A. Agarwal, B. Furht, M. Conatser, and C. Baechle, "Secure Mobile Framework for Monitoring Medical Sensor Data," in *Handbook of Medical and Healthcare Technologies*, Springer, 2013, pp. 355–369.

[34] A. Smola and B. Schölkopf, "A tutorial on support vector regression," *Stat. Comput.*, vol. 14, pp. 199–222, 2004.

[35] R. Amarasingham, B. J. Moore, Y. P. Tabak, M. H. Drazner, C. A. Clark, S. Zhang, W. G. Reed, T. S. Swanson, Y. Ma, E. A. Halm, R. Amarasingham MD, MBA, B. J. Moore PhD, Y. P. Tabak PhD, M. H. Drazner MD, MSc, C. A. Clark MPA, S. Zhang PhD, W. G. Reed MD, T. S. Swanson BA, Y. Ma PhD, and E. A. Halm MD, MPH, "An automated model to identify heart failure patients at risk for 30-day readmission or death using electronic medical record data," *Med. Care*, vol. 48, no. 11, pp. 981–988, 2010.

[36] H. Krumholz, S. L. Normand, P. Keenan, Z. Lin, E. E. Drye, K. R. Bhat, Y. Wang, J. S. Ross, J. D. Schuur, B. Stauffer, and others, "Hospital 30-day heart failure readmission measure methodology," *Rep. Prep. Centers Medicare Medicaid Serv.*, 2008.

[37] N. Wettersten, M. Wilson, K. Tong, and J. E. López, "Enhanced Prediction of Heart Failure 30-Day Readmission Risk With Four Readily-Available Clinical Variables," *Circulation*, vol. 130, no. Suppl 2, pp. A17063--A17063, 2014.

[38] V. Betihavas, S. A. Frost, P. J. Newton, P. Macdonald, S. Stewart, M. J. Carrington, Y. K. Chan, and P. M. Davidson, "An Absolute Risk Prediction Model to Determine Unplanned Cardiovascular Readmissions for Adults with Chronic Heart Failure," *Hear. Lung Circ.*, vol. 24, no. 11, pp. 1068–1073, 2015.

[39] P. Thavendiranathan, T. Yingchoncharoen, A. Grant, S. Seicean, S. H. Landers, E. Z. Gorodeski, and T. H. Marwick, "Prediction of 30-day heart failure-specific readmission risk by echocardiographic parameters," *Am. J. Cardiol.*, vol. 113, no. 2, pp. 335–341, 2014.

[40] AHRQ, "HCUP STATISTICAL BRIEF #153," 2013. [Online]. Available: https://www.hcup-us.ahrq.gov/reports/statbriefs/sb153.pdf.

[41] A. Agarwal, R. S. Behara, S. Mulpura, and V. Tyagi, "Domain Independent Natural Language Processing -- A Case Study for Hospital Readmission with COPD," *2014 IEEE*

Int. Conf. Bioinforma. Bioeng., pp. 399–404, 2014.

[42] R. P. Merkow, M. H. Ju, J. W. Chung, B. L. Hall, M. E. Cohen, M. V Williams, T. C. Tsai, C. Y. Ko, and K. Y. Bilimoria, "Underlying reasons associated with hospital readmission following surgery in the United States," *Jama*, vol. 313, no. 5, pp. 483–495, 2015.

[43] E. H. Lawson, B. L. Hall, R. Louie, S. L. Ettner, D. S. Zingmond, L. Han, M. Rapp, and C. Y. Ko, "Association Between Occurrence of a Postoperative Complication and Readmission," *Ann. Surg.*, vol. 258, no. 1, pp. 10–18, 2013.

[44] D. E. Fry, M. Pine, D. Locke, and G. Pine, "Prediction models of Medicare 90-day postdischarge deaths, readmissions, and costs in bowel operations," *Am. J. Surg.*, vol. 209, no. 3, pp. 509–514, 2015.

[45] R. P. Kiran, C. P. Delaney, A. J. Senagore, M. Steel, T. Garafalo, and V. W. Fazio, "Outcomes and prediction of hospital readmission after intestinal surgery," *J. Am. Coll. Surg.*, vol. 198, no. 6, pp. 877–883, 2004.

[46] C. Baechle, A. Agarwal, R. Behara, and X. Zhu, "A Cost Sensitive Approach to Predicting 30-Day Hospital Readmission in COPD Patients," in *Biomedical and Health Informatics (BHI), 2017 IEEE-EMBS International Conference on*, 2017, pp. 317–320.

[47] K. Carey and T. Stefos, "The cost of hospital readmissions: evidence from the VA," *Health Care Manag. Sci.*, vol. 19, no. 3, pp. 241–248, 2016.

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

[49] J. Ramos, J. Eden, and R. Edu, “Using TF-IDF to Determine Word Relevance in Document Queries,” *Processing*, 2003.

[50] S. T. Wu, H. Liu, D. Li, C. Tao, M. A. Musen, C. G. Chute, and N. H. Shah, “Unified Medical Language System term occurrences in clinical notes: a large-scale corpus analysis.,” *J. Am. Med. Inform. Assoc.*, vol. 19, no. e1, pp. e149-56, 2012.

[51] C. Baechle, A. Agarwal, R. Behara, and X. Zhu, “Co-Occurring Evidence Discovery for COPD Patients using Natural Language Processing,” pp. 321–324, 2017.

[52] M. Herland, T. M. Khoshgoftaar, and R. Wald, “A review of data mining using big data in health informatics,” *J. Big Data*, vol. 1, no. 1, p. 2, 2014.

[53] C. Baechle, A. Agarwal, and X. Zhu, “Big data driven co-occurring evidence discovery in chronic obstructive pulmonary disease patients,” *J. Big Data*, vol. 4, no. 1, p. 9, 2017.

[54] K. S. Jones, “Natural languages processing: a historical review,” *Univ. Cambridge*, vol. 7, pp. 2–10, 2001.

[55] S. Bird, E. Klein, and E. Loper, “NLTK Book.” Sebastopol, CA: O’Reilly Media, 2009.

[56] C. D. Manning, P. Raghavan, H. Schütze, and others, *Introduction to information retrieval*, vol. 1, no. 1. Cambridge university press Cambridge, 2008.

[57] C. Trim, “The Art of Tokenization,” *IBM Developerworks*, Jan-2013. [Online]. Available: https://www.ibm.com/developerworks/community/blogs/nlp/entry/tokenization. [Accessed: 06-Jul-2016].

[58] J. O’neil, “Doing Things with Words: Sentence Boundary Detection,” *Attivio*, Jan-

2008. [Online]. Available: http://bit.ly/1UdkAKn. [Accessed: 06-Jul-2016].

[59] J. H. Martin and D. Jurafsky, “Speech and language processing,” *Int. Ed.*, 2000.

[60] M. F. Porter, “An algorithm for suffix stripping,” *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[61] D. Porter, “The Porter Stemming Algorithm,” Jan-2006. [Online]. Available: http://tartarus.org/martin/PorterStemmer/. [Accessed: 06-Jul-2016].

[62] R. M. Reese, Natural Language Processing with Java (Community Experience Distilled). Packt Publishing, 2015.

[63] C. Manning, “Lecture notes.” Apr-2012.

[64] “About Grok.” [Online]. Available: https://wiki.apache.org/incubator/OpenNLPProposal. [Accessed: 06-Jul-2016].

[65] G. S. Ingersoll, T. S. Morton, and A. L. Farris, *Taming text: how to find, organize, and manipulate it*. Manning Publications Co., 2013.

[66] C. D. Manning, J. Bauer, J. Finkel, S. J. Bethard, M. Surdeanu, and D. McClosky, “The Stanford CoreNLP Natural Language Processing Toolkit,” *Proc. 52nd Annu. Meet. Assoc. Comput. Linguist. Syst. Demonstr.*, pp. 55–60, 2014.

[67] M. Fowler, “Inversion of control containers and the dependency injection pattern.” 2004.

[68] S. Bethard, P. V Ogren, and L. Becker, “ClearTK 2.0: Design Patterns for Machine Learning in UIMA.,” in *LREC*, 2014, pp. 3289–3293.

[69] D. Ferrucci and A. Lally, “UIMA: an architectural approach to unstructured information processing in the corporate research environment,” *Nat. Lang. Eng.*, vol. 10, no. 3–4, pp. 327–348, 2004.

[70] “Getting Started: Apache UIMA C++ Framework.” [Online]. Available: https://uima.apache.org/doc-uimacpp-huh.html. [Accessed: 07-Jun-2016].

[71] “Apache UIMA Project Incubation Status.” [Online]. Available: https://incubator.apache.org/projects/uima.html. [Accessed: 07-Jun-2016].

[72] P. Ogren and S. Bethard, “Building Test Suites for UIMA Components,” in Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing (SETQA-NLP 2009), 2009, pp. 1–4.

[73] H. Cunningham, D. Maynard, K. Bontcheva, V. Tablan, N. Aswani, I. Roberts, G. Gorrell, A. Funk, A. Roberts, D. Damljanovic, T. Heitz, M. A. Greenwood, H. Saggion, J. Petrak, Y. Li, and W. Peters, *Text Processing with GATE (Version 6)*. 2011.

[74] S. Grimes, “Unstructured Data and the 80% Rule,” 2008. [Online]. Available: https://breakthroughanalysis.com/2008/08/01/unstructured-data-and-the-80-percent-rule/. [Accessed: 07-Jun-2016].

[75] A. R. Aronson, “Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program.,” in *Proceedings of the AMIA Symposium*, 2001, p. 17.

[76] O. Bodenreider, “The unified medical language system (UMLS): integrating biomedical terminology,” *Nucleic Acids Res.*, vol. 32, no. suppl 1, pp. D267--D270, 2004.

[77] D. A. B. Lindberg, B. L. Humphreys, A. T. McCray, and others, “The unified medical language system,” *IMIA Yearb.*, pp. 41–51, 1993.

[78] P. Ruch, J. Gobeill, C. Lovis, and A. Geissbühler, “Automatic medical encoding with SNOMED categories.,” *BMC Med. Inform. Decis. Mak.*, vol. 8 Suppl 1, p. S6, 2008.

[79] N. Sioutos, S. de Coronado, M. W. Haber, F. W. Hartel, W. L. Shaiu, and L. W. Wright, “NCI Thesaurus: A semantic model integrating cancer-related clinical and

molecular information,” *J. Biomed. Inform.*, vol. 40, no. 1, pp. 30–43, 2007.

[80] C. E. Lipscomb, “Medical Subject Headings (MeSH).,” *Bull. Med. Libr. Assoc.*, vol. 88, no. 3, pp. 265–6, 2000.

[81] United States Department of Health and Human Services, *The International Classification of Diseases*. World Health Organization, 1969.

[82] V. N. Slee, “The International classification of diseases: ninth revision (ICD-9),” *Ann. Intern. Med.*, vol. 88, no. 3, pp. 424–426, 1978.

[83] “International Classification of Diseases,Ninth Revision (ICD-9).” [Online]. Available: http://www.cdc.gov/nchs/icd/icd9.htm. [Accessed: 11-Jul-2016].

[84] N. Sager, *Natural language information processing*. Addison-Wesley Publishing Company, Advanced Book Program, 1981.

[85] C. Friedman, “A broad-coverage natural language processing system.,” *AMIA Annu. Symp. Proc.*, pp. 270–4, 2000.

[86] “Columbia Grants Health Fidelity Exclusive License to MedLEE NLP,” 2012. [Online]. Available: http://www.businesswire.com/news/home/20120111006135/en/Columbia-Grants-Health-Fidelity-Exclusive-License-MedLEE. [Accessed: 07-Jun-2016].

[87] G. K. Savova, J. J. Masanz, P. V Ogren, J. Zheng, S. Sohn, K. C. Kipper-Schuler, and C. G. Chute, “Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications,” *J. Am. Med. Informatics Assoc.*, vol. 17, no. 5, pp. 507–513, 2010.

[88] C. Baechle, A. Agarwal, R. S. Behara, and X. Zhu, “Latent Topic Ensemble Learning for Hospital Readmission Cost Reduction,” in *2017 International Joint*

Conference on Neural Networks (IJCNN), 2017.

[89] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, no. Jan, pp. 993–1022, 2003.

[90] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 193–200.

[91] United States Department of Health and Human Services, "Summary of the HIPAA privacy rule," *Washington, DC Dep. Heal. Hum. Serv.*, 2003.

[92] D. D. Lewis, "Reuters-21578 text categorization collection data set," 1997)[2012--05--30]. http://archive. ics. uci. edu/ml/datasets/Reuters21578+ Text+ Categorization+ collection. 1998.

[93] J. McAuley and J. Leskovec, "Hidden factors and hidden topics: understanding rating dimensions with review text," *Proc. 7th ACM Conf. Recomm. Syst. - RecSys '13*, pp. 165–172, 2013.

[94] Agency for Healthcare Research and Quality, "HCUP NRD Overview." [Online]. Available: https://www.hcup-us.ahrq.gov/nrdoverview.jsp.

[95] Agency for Healthcare Research and Quality, "HCUP Overview." [Online]. Available: https://www.hcup-us.ahrq.gov/overview.jsp.

[96] T. L. Petty, "The history of COPD Early historical landmarks," *Int. J. COPD*, vol. 1, pp. 3–14, 2006.

[97] A. Marengoni, D. Rizzuto, H. X. Wang, B. Winblad, and L. Fratiglioni, "Patterns of chronic multimorbidity in the elderly population," *J. Am. Geriatr. Soc.*, vol. 57, no. 2, pp. 225–230, 2009.

[98] C. P. Aaron, J. E. Schwartz, E. A. Hoffman, R. Tracy, J. H. M. Austin, L. J. Smith,

D. R. Jacobs, K. E. Watson, and R. G. Barr, “Aspirin Use And Longitudinal Progression Of Percent Emphysema On CT : The MESA Lung Study,” vol. 4, p. 100543, 2015.

[99] W. D’Hoore, C. Sicotte, and C. Tilquin, “Risk adjustment in outcome assessment: the Charlson comorbidity index.,” *Methods Inf. Med.*, vol. 32, no. 5, pp. 382–387, 1993.

[100] D. Demner-Fushman, W. W. Chapman, and C. J. McDonald, “What can natural language processing do for clinical decision support?,” *J. Biomed. Inform.*, vol. 42, no. 5, pp. 760–772, 2009.

[101] Y. Wu, J. C. Denny, S. T. Rosenbloom, R. A. Miller, D. A. Giuse, and H. Xu, “A comparative study of current Clinical Natural Language Processing systems on handling abbreviations in discharge summaries.,” *AMIA Annu. Symp. Proc.*, vol. 2012, pp. 997–1003, 2012.

[102] WebMD, “COPD Comorbid Conditions: Heart Disease, Osteoporosis, and More.” [Online]. Available: http://wb.md/2dGwUqq. [Accessed: 01-Aug-2016].

[103] CDC, “Addressing the Nation’s Most Common Cause of Disability At A Glance 2015,” 2015. [Online]. Available: http://bit.ly/1FKbR7i. [Accessed: 01-Aug-2016].