This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2017.2684121, IEEE Journal of Biomedical and Health Informatics

A Natural Language Processing Framework for Assessing Hospital Readmissions for Patients with COPD

Ankur Agarwal[1], Christopher Baechle[1], Ravi Behara[2], Xingquan Zhu[1]

[1] Dept. of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431, USA
[2]Dept. of IT & Operations Management, Florida Atlantic University, Boca Raton, FL 33431, USA

Abstract—**With the passage of recent federal legislation many medical institutions are now responsible for reaching target hospital readmission rates. Chronic diseases account for many hospital readmissions and Chronic Obstructive Pulmonary Disease has been recently added to the list of diseases for which the United States government penalizes hospitals incurring excessive readmissions. Though there have been efforts to statistically predict those most in danger of readmission, few have focused primarily on unstructured clinical notes. We have proposed a framework which uses Natural Language Processing to analyze clinical notes and predict readmission. Many algorithms within the field of data mining and machine learning exist, so a framework for component selection is created to select the best components. Naïve Bayes using Chi-Squared feature selection offers an AUC of 0.690 while maintaining fast computational times.**

Keywords—Natural language processing, Medical information systems, Decision support systems, Data mining, Feature Extraction

I. Introduction

The American Recovery and Reinvestment Act (ARRA) of 2009 [1] emphasized the adoption of health information technology through the Health Information Technology for Economic and Clinical Health Act (HITECH Act) [2]. Two prime components related to this act are: (1) Introduction of penalties for hospitals for patient readmission within 30, 60 and 90 day period for specific diagnoses; (2) Introduction of the concept of Clinical Decision Support Systems (CDSS) in Electronic Health Records through “Meaningful Use” (MU) compliance [3]. Currently, the MU compliance requires a very basic implementation of rule based decision support systems which could be introduced by an office-practice physician based on the combination of demographics, lab results, medications, allergy, and past medical history.

The HITECH Act stipulates that healthcare providers demonstrate the meaningful use of health IT. As part of this act, CMS identified “hospital readmissions for COPD” as a costly problem that needs to be addressed in the United States as a whole [4]. The scope of the problem is very large and cost data is available through CMS. CMS has started penalizing hospitals for excessive 30-day COPD readmissions. As a result, there is an increased amount of pressure on hospitals to adopt the CDSS to identify the candidates for hospital readmission and avoid such readmissions by a series of efforts, such as closely coordinated transition of care. Unfortunately, it is not possible to provide such an extensive level of care for every patient due to the amount of resources needed, shortage in medical staff, and the expenses involved in such care coordination [4]–[6]. Therefore, it is critical to accurately identify candidates for hospital readmission and then avoid such readmission through the use of resources. Further, since patient-hospitalization represents such a large portion of healthcare expenses, health plans, Accountable Care Organizations (ACO), and Managed Services Organizations (MSO) are also targeting hospital readmission in order to improve their profitability. Though predictive modeling for many diseases have seen a large body of research [7]–[10], COPD predictive modeling remains scarce.

Patient data in hospitals includes a significant amount of unstructured data. Examples include physician’s notes, discharge summaries, and x-ray radiology reports. Since free text is an important part of patient records, including it in predictive analysis is equally important. Despite the inherent value of the clinical information present in the document, a manual review of free text records is very time-consuming process. Therefore, there is interest in developing a Natural Language Processing (NLP) based approach to extract such information from patient records. However, this is not a simple task due to the ambiguity and variations in language used for describing and evaluating any specific patient condition. User specific use of terminology, abbreviations, and acronyms are often used for describing patient condition. Every physician has a unique style and terminologies for defining a patient problem, encounter or a situation. Due to the variation and complexity in such unstructured information, an architecture which can standardize the information by converting this unstructured data into structured form is required.

II. Backgound

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 are not implemented from scratch each time a system is developed. These software components fall primarily into two categories: libraries and frameworks.

A. *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 meant to enable programmers to write software without duplication of efforts. NLP has many libraries available, aimed at different languages and purposes. This research makes use of OpenNLP, a Java based NLP library.

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 [11]. 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 [12]. 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. Training these models however requires documents be annotated manually as most of the utilized 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.

B. *NLP Frameworks*

Often times different NLP systems have very similar designs. Higher level tasks commonly depend upon lower level tasks. 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 the two is frameworks rely on Inversion of Control (IoC) [13]. 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 [14]. 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 part-of-speech tagging, then stemming, with each task depending upon information from the previous task. 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. This research makes use of Apache Unstructured Information Architecture (UIMA).

Apache UIMA is a framework that started at IBM research in 2004 to address the growing need to structure large systems that processed unstructured data [15]. 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 [16]. 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.

C. *Medical NLP*

The medical domain has been one of the earliest applications of NLP [17]–[19]. 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 [20]. 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.

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 Apache UIMA [21].

cTAKES uses Commercial Off The Shelf (COTS) software components for many parts of the system. Apache OpenNLP provides 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 Fig. 1.

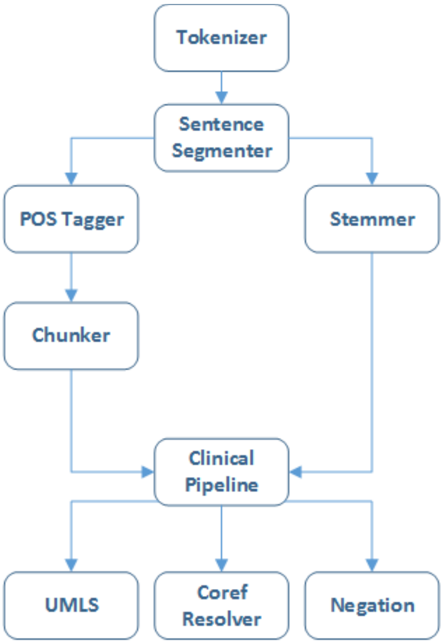

Fig. 1 cTAKES components

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.

III. Related Works

With the passage of federal legislation penalizing excessive 30 day hospital readmission, readmission risk modeling has been an active area of research. A systematic review was performed in 2011 by Kansagara et al. which compares data, methodology, and results [22]. The review confirmed that readmission prediction is a difficult problem and recent models do not necessarily perform better than research a decade prior. Research which attempted to solve the 30-day readmission problem generally performed worse than those trying to solve many-months readmission. A possible reason for this is more patients are readmitted for long periods of time, thus creating slightly more balanced class distribution. Additionally, many systems were able to increase performance by looking at a specific subset of patients, such as those with Congestive Heart Failure (CHF).

The sample size of patients varied greatly. A study using England's National Health Service (NHS) analyzed over 1.4 million patients (AUC=0.72) [23] while another study only used 487 patients (AUC=0.70) [24]. There seems to be little correlation to dataset size and model performance for this problem. The study which claimed the best performance (AUC=0.83) only used 700 patients [25]. Unsurprisingly, quality of data tended to have a large impact upon model quality and models using data from Centers for Medicare and Medicaid Services (CMS) tended to perform well. Very few models were able to perform better than an AUC=0.68.

Models generally used structured data including medical diagnosis and severity, history of condition, overall health of patient, and sociodemographic features. Surprisingly, none of the studies analyzed used NLP to analyze clinical notes, nor did any of the studies subset COPD patients. While newer research has made some use of clinical notes to model readmission, research in this area is scarce.

1) Prediction of Hospital Readmission in COPD Patients

A framework was created by researchers at Deakin University to analyze many chronic disease readmissions [26]. Diseases are matched to templates of features and COPD is a disease that was studied. This system represents one of the few research efforts to analyze COPD patients (though does not focus solely on COPD) and additionally focuses specifically on the 30-day readmission classification task. The system works by creating schemas to be used when capturing data for a patient. In the case of COPD, a COPD specific template is used. Disease specific models are then built. This method works well because as previously mentioned, building a single classifier for a diverse patient population can often decrease model performance. 1,816 patients were analyzed. The model is able to predict 30-day readmission rate in COPD patients with an AUC=0.67 and was an improvement upon co-morbidity baseline methods that are often used for readmission analysis.

Another system which is specific to COPD patients was created by Fan et al. [27]. This system was not however used in the analysis of hospital readmissions. Instead, patients were analyzed for COPD exacerbations within the period of one year. Baseline methods for comparison used a model consisting of basic features such as demographics and questionnaire information. An improved model was presented which also included the features spirometry, PaO2, dyspnea, prior exacerbations and co-morbidity. The AUC for this model was 0.68. Though the model was not specific to hospital readmissions, it is not unreasonable to infer that many of these exacerbations would have resulted in a hospital visit.

2) Prediction of 30 Day Readmission using EHR

Work by Wasfy et al. attempts to use the unstructured data contained in the Electronic Health Record (EHR) to predict 30 day readmission in percutaneous coronary intervention patients [10]. The primary method of NLP in this study was the use of regular expressions to extract specific queries from the clinical note. This method is in opposition to the so-called *bag-of-words* representation of features which treats each word as a feature and requires no domain knowledge. Although using regular expression based queries can be useful, automatic discovery of new features which may be useful is not possible. The AUC for this research was 0.69. The data distribution was changed to approximately 0.33 readmitted and 0.67 not readmitted. There is no mention of a separate test dataset or cross validation methods, thus it is difficult to infer if the AUC value was calculated on a re-balanced dataset or a pristine held-out dataset.

3) *Prediction of 30 day readmission using EHR using Apache cTAKES*

Recent research by Duggal et al. uses Apache cTAKES to annotate the unstructured EHR [28]. This research specifically looks at the 30-day readmission rate of diabetes patients in an Indian hospital. The data contains a .129 readmission rate and 9,381 instances. Several machine learning algorithms were compared. Naïve Bayes, Logistic Regression, Random Forest, Adaboost, and Neural Networks. The highest AUC for this diabetes study was 0.688 using Random Forests. The results are typical of readmission analysis.

No run time analysis was performed for this research. Few published works on readmission include this metric, but in practice is extremely important. Random Forest is a computationally intense model which may take orders of magnitude longer to build than simple methods like Naïve Bayes. Additionally, basic feature analysis was performed. However, rigorous analysis of feature selection techniques and their effects on model performance would be preferable.

Direct comparison of methodologies in the field of health informatics can be difficult. Unlike other domains, data is often restricted and cannot be released publicly. The work by Duggal et al. represents the only other work we were able to find which uses a similar methodology and has only been very recently published. Since the data is unavailable, we compare a baseline method and compare proposed improvements using a mix of bag-of-words methodology and cTAKES annotations using feature selection methods. Though research using NLP to predict hospital readmissions has begun to appear in the last two years, no research has explored feature selection techniques for purposes of readmission, and limited publications have been done in exploring COPD patient readmission, regardless of methodology [29].

IV. Methodology

The framework can be broken down into four subsystems: (1) Feature extraction (2) Feature selection (3) Classification (4) Performance evaluation. Fig. 2 outlines these subsystems.

A. *Data*

The data used to test this system 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% true and 85.7% false. The data evaluated using n-fold cross validation where n=10.

B. *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. Our model has two distinct phases of feature engineering, feature extraction and feature selection. Given a piece of natural language, several methods exist for extracting features.

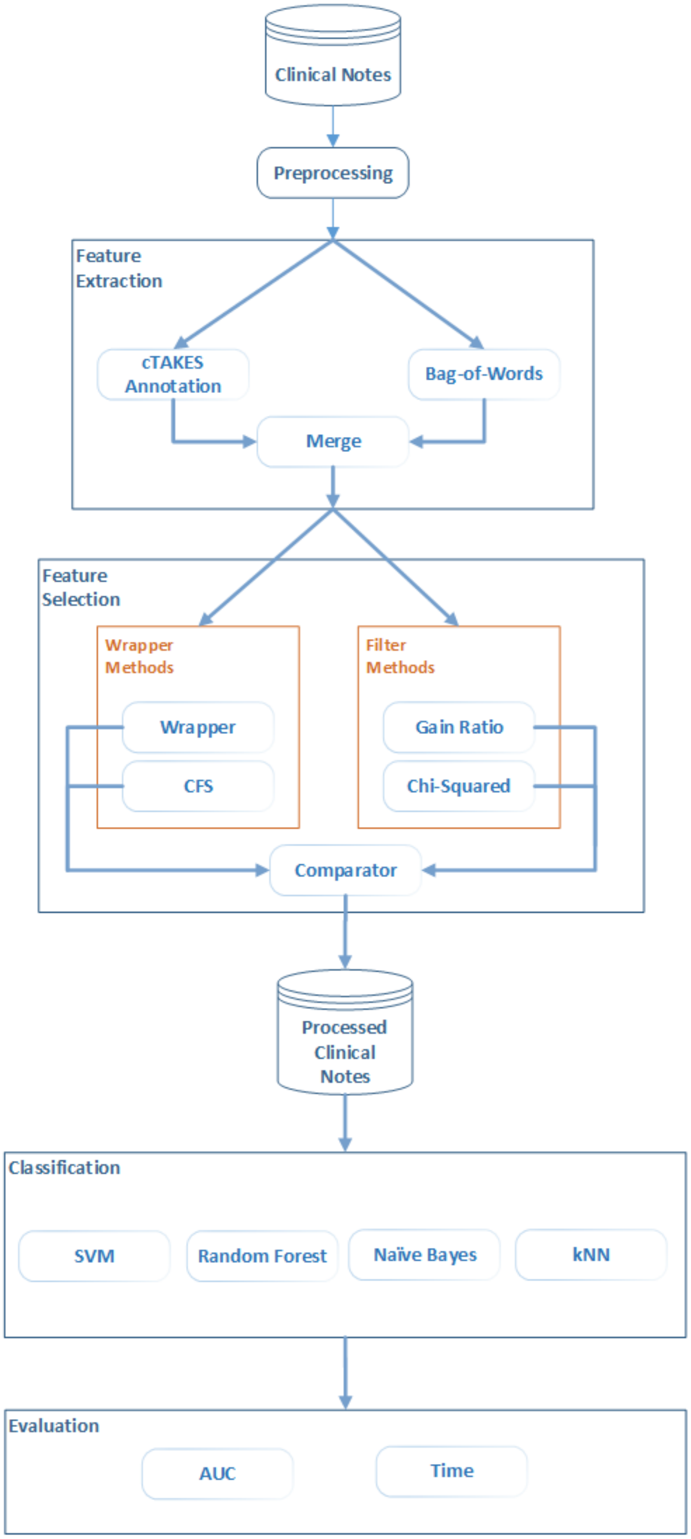

Fig. 2 Block diagram of framework.

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.

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.

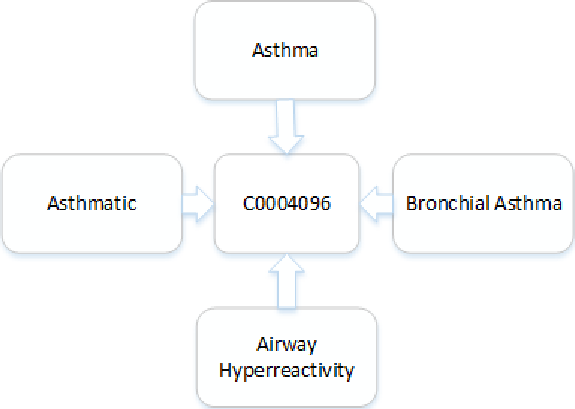

Fig. 3 Various forms of the CID C0004096 representing asthma.

This higher level feature extraction is done using Apache cTAKES. Diseases and disorders are extracted and normalized using UMLS to a single CID. Fig. 3 shows an example of a disease with several representations and its normalized form. Medications are additionally extracted and normalized to their common name. Both bag-of-words and cTAKES annotations are used in this framework and the resultant set of features is the union of sets.

C. 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.

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.

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) *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.

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.

D. *Classification*

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.

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) *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.

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.

E. *Evaluation*

1) *Area Under Receiver Operating Characteristic Curve*

The COPD readmission data has a large class imbalance. The distribution is .143 is true for readmission within 30 days, and .857 is false. Accuracy is often an intuitive measure for classification performance but can be misleading. Creating a classifier which classifies all instances of this dataset as false will have an accuracy of .857, but will have no useful predictive power. In datasets containing class imbalance, the Area Under the Receiver Operating Characteristic (AUC) is often more useful in evaluating model performance.

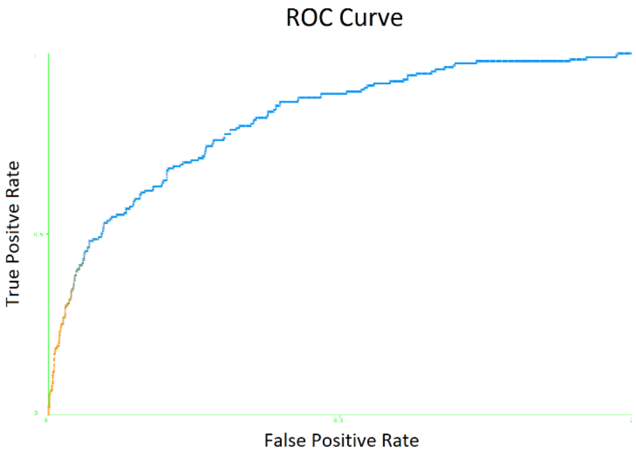

Fig. 4 Example ROC Curve.

The Receiver Operating Characteristic (ROC) is a measure of the True Positive Rate (TPR) against False Positive Rate (FPR). Most machine learning algorithms can be coerced to output a probability of classification for each test instance. By using these class probabilities and varying the threshold of classification a curve can be plotted. The area under this curve is known as AUC and the larger the area, the better a classifier can have a large TPR without increasing the FPR.

2) *Time*

Model creation time and classification of instances is often important. Some algorithms may take orders of magnitude longer to create than the fastest algorithms and only provide a small increase in classification performance. NB is known to be very fast because the algorithm simply does frequency counts of feature values for each instance. Algorithms such as RF which use many classifiers to arrive at a classification are known to be much slower. Each algorithm is evaluated 10 times and an average time is recorded.

Feature selection algorithms can additionally vary widely in execution time. Statistical methods such as GR and CS are typically much faster than methods such as wrapper. Wrapper must iteratively perform an evaluation of the features selected to determine an optimal subset and can take orders of magnitude longer to complete than statistical methods. Each feature selection algorithm is evaluated 10 times and an average time is recorded.

F. *Weka*

The machine learning toolkit used in this framework is the Waikato Environment for Knowledge Analysis (Weka) [30]. Weka contains implementations of the machine learning and feature selection algorithms used in this research. The Weka Java API is used to write the framework. The software versions used are 3.6.14 for Weka and Java JDK version 8u91.

V. RESULTS

A. *Feature Selection*

TABLE I. SELECTION OF FEATURES DISCOVERED BY FORWARD SELECTION WRAPPER METHOD IN AT LEAST 9 OF 10 FOLDS.

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 II. SELECTION OF FEATURES DISCOVERED BY CFS METHOD IN AT LEAST 9 OF 10 FOLDS.

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 III. HIGHEST SCORING FEATURES ON THE FULL DATASET FOR GR FEATURE SELECTION.

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 IV. HIGHEST SCORING FEATURES ON THE FULL DATASET FOR CS FEATURE SELECTION.

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.

Tables I-IV show a selection of the top features found for each feature selection algorithm. Unlike GR and CS, wrapper and CFS methods choose an optimal subset of features. Table V 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 resulting in $p \leq 0.01$. Thus, the two methods found a statistically significant different number of features to be optimal.

TABLE V. COMPARISON OF NUMBER OF FEATURES FOUND TO BE OPTIMAL BY WRAPPER AND CFS FEATURE SELECTION METHODS.

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

An analysis of intersecting features was performed and Table VI 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.

TABLE VI. OVERLAP OF FEATURES BETWEEN FEATURE SELECTORS.

	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

B. *AUC*

An analysis of the AUC of classifiers was performed and presented in Table VII. 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.

TABLE VII. AUC OF CLASSIFIERS, GROUPED BY FEATURE SELECTOR. STATISTICAL SIGNIFICANCE TESTED AGAINST BASELINE METHOD OF NO FEATURE SELECTION.

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	
***Results are statistically significant at $p \leq 0.05$**					

Improvement of models over the baseline method (no feature selection) is seen. AUC improvement is analyzed using a paired t-test against the baseline, with each fold representing an AUC value and a threshold of $p \leq 0.05$ 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 over baseline methodology. 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.

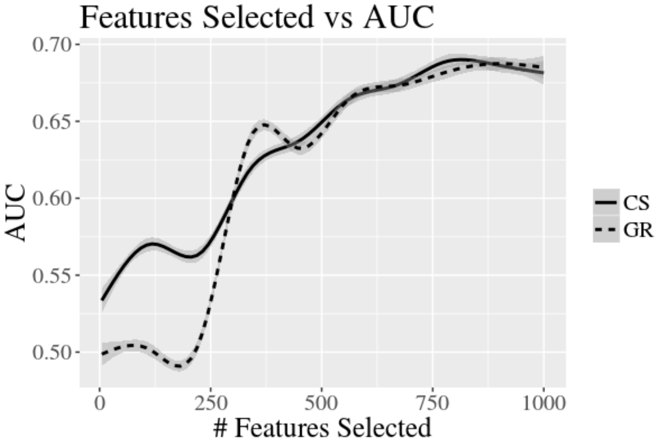

Fig. 5 Effect upon AUC when varying # features for CS, averaged over 10 folds. Standard error is outlined in gray.

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 Fig. 5), 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.

C. *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 VIII 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 IX 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.

TABLE VIII MODEL CREATION AND EVALUATION TIME (MS). AVERAGED OVER 10 FOLDS.

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 IX. FEATURE SELECTION ALGORITHM TIME. AVERAGED OVER 10 FOLDS.

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

VI. Conclusion

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. Future work intents to extend efforts to other chronic diseases as records become available.

VII. Acknowledgement

This project was funded by the National Science Foundation under grant numbers IIP-1230693 and IIP-1444949.

VIII. References

[1] "The American Recovery and Reinvestment Act of 2009 Report."

[Online]. Available: http://www.gpo.gov/fdsys/pkg/BILLS-111hr1enr/pdf/BILLS-111hr1enr.pdf. [Accessed: 23-Aug-2016].

[2] "HITECH Act Enforcement Interim Final Rule." [Online]. Available: http://www.hhs.gov/ocr/privacy/hipaa/administrative/enforcementrule/hitechenforcementifr.html. [Accessed: 23-Aug-2016].

[3] "Electronic Health Records (EHR) Incentive Programs." [Online]. Available: https://www.cms.gov/Regulations-and-Guidance/Legislation/EHRIncentivePrograms/index.html. [Accessed: 23-Aug-2016].

[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] P. Jain, *Prognostic COPD healthcare management system*, no. May. FLORIDA ATLANTIC UNIVERSITY, 2014.

[6] R. Behara, A. Agarwal, F. Fatteh, and B. Furht, "Predicting Hospital Readmission Risk for COPD Using EHR Information," in *Handbook of Medical and Healthcare Technologies*, Springer, 2013, pp. 297–308.

[7] R. Behara, A. Agarwal, V. Rao, and C. Baechle, "Predicting the Occurrence of Diabetes using Analytics," in *Models and Applications in the Decision Sciences: Best Papers from the 2015 Annual Conference*, 1st ed., Pearson Press, 2016, pp. 187–193.

[8] R. Behara, A. Agarwal, V. Rao, and C. Baechle, "Predictive Analytics for Chronic Diabetes Care," in *2015 Annual Meeting of the Decision Sciences Institute Proceedings*, 2015.

[9] R. Wallmann, J. Llorca, I. Gómez-Acebo, Á. C. Ortega, F. R. Roldan, and T. Dierssen-Sotos, "Prediction of 30-day cardiac-related-emergency-readmissions using simple administrative hospital data," *Int. J. Cardiol.*, vol. 164, no. 2, pp. 193–200, 2013.

[10] J. H. Wasfy, G. Singal, C. O'Brien, D. M. Blumenthal, K. F. Kennedy, J. B. Strom, J. A. Spertus, L. Mauri, S. L. T. Normand, and R. W. Yeh, "Enhancing the Prediction of 30-Day Readmission after Percutaneous Coronary Intervention Using Data Extracted by Querying of the Electronic Health Record," *Circ. Cardiovasc. Qual. Outcomes*, vol. 8, no. 5, pp. 477–485, 2015.

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

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

[13] M. Fowler, "Inversion of control containers and the dependency injection pattern." 2004.

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

[15] 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.

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

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

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

[19] 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.

[20] 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].

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] T. Tran, W. Luo, D. Phung, S. Gupta, S. Rana, R. Kennedy, A. Larkins, and S. Venkatesh, "A framework for feature extraction from hospital medical data with applications in risk prediction.," *BMC Bioinformatics*, vol. 15, no. 1, p. 6596, 2014.

[27] V. S. Fan, S. D. Ramsey, B. J. Make, and F. J. Martinez, "Physiologic variables and functional status independently predict COPD hospitalizations and emergency department visits in patients with severe COPD," *COPD J. Chronic Obstr. Pulm. Dis.*, vol. 4, no. 1, pp. 29–39, 2007.

[28] 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.

[29] 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.

[30] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.