Automated Mapping of Laboratory Tests to LOINC Codes using Noisy Labels in a National Electronic Health Record System Database

By

Sharidan Kristen Parr

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

in

Biomedical Informatics

August 10, 2018

Nashville, Tennessee

Approved

Michael E. Matheny, M.D., M.S., M.P.H.

Thomas A. Lasko, M.D., Ph.D.

Matthew S. Shotwell, Ph.D.

ACKNOWLEDGEMENTS

This work was made possible by the financial support of the Department of Veterans Affairs, Office of Academic Affiliations, Advanced Fellowship Program in Medical Informatics; the Vanderbilt University Medical Center's Department of Biomedical Informatics; and the Veterans Health Administration Health Services Research & Development (HSR&D) Investigator Initiated Research grant IIR 13-052.

I am grateful for the mentorship, direction, and support provided by my Master's thesis committee members, Drs. Michael Matheny, Thomas Lasko, and Matthew Shotwell. As my primary advisor over the past three years, Dr. Matheny stimulated my intellectual curiosity, challenged, and encouraged me through completion of the research project, while providing instrumental guidance for my career development. Drs. Lasko and Shotwell were engaged collaborators, enthusiastically providing direction and critical scientific perspective throughout the project. Collectively, my committee members created an environment that facilitated the successful completion of my successful Master's thesis research.

I was also privileged to work with numerous collaborators at both the U.S. Department of Veterans Affairs and Vanderbilt's Department of Biomedical Informatics. Jejo Koola, Ruth Reeves, and Glenn Gobbel provided encouragement and insight for the conceptual design and feature generation. Fern Fitzhenry provided guidance with respect to LOINC resources. Jason Denton provided technical support for complicated database queries. Dax Westerman assisted with securing and maintaining computational resources. Alvin Jeffery aided with manual adjudication. I greatly appreciate the assistance and support they provided.

TABLE OF CONTENTS

Page

Chapter

LIST OF TABLES

LIST OF FIGURES

Figure Page

LIST OF ABBREVIATIONS

API – Application Program Interface

CPRS – Computerized Patient Record System

CUI - Concept Unique Identifier

EHR - Electronic Health Record

HITCH - Health Information Technology for Economic and Clinical Health

LOINC - Logical Observation Identifiers Names and Codes

RELMA – Regenstrief LOINC Mapping Assistant

REST – Representational State Transfer

UMLS - Unified Medical Language System

VA – Veterans Affairs

VistA – Veterans Information Systems and Technology Architecture

CHAPTER 1

INTRODUCTION AND BACKGROUND

Introduction

Clinical data, information collected during the course of patient care, is essential for describing and monitoring a patient's state of health. Examples of clinical data include laboratory test results, vital signs (e.g. temperature or blood pressure), medical diagnoses (e.g. diabetes), radiology test results (e.g. chest X-rays), and prescription drug information. For clinical data to be effectively stored and communicated, data elements must be mapped to a standardized terminology,[1,2] a common language that is shared among users. This common language links clinical data elements with a standardized coding and classification system.[3]

Aggregate data sources containing clinical data from multiple healthcare sites are valuable for research, quality, public health, and creating large evidence bases to answer clinical questions.[4,5] Common data models, which standardize the format and content of observational data, hold promise for integrating disparate data sources in healthcare. A key requirement in this process is that institution-specific information must be mapped to a standardized terminology. Without this mapping, clinical data cannot be combined, shared, or interpreted in a meaningful context.[1,2]

Laboratory tests, collected from a patient's blood, urine, or other body tissue provide information about a patient's health in order to prevent, diagnose, and treat disease. The standard code system for laboratory observations, the Logical Observation Identifiers Names and Codes (LOINC®),[6] aims to facilitate data aggregation for quality measures, outcomes research, and health information exchange.[7] Historically, electronic health record (EHR) implementations have used proprietary data mapping with locally-defined, idiosyncratic, ambiguous identifiers[8] that make mapping to standard terminologies challenging. Furthermore, even when LOINC codes are used, they are often incorrectly mapped.[9] As a result, accurately mapping these data to standards for incorporation into common data models is time-consuming and resource intensive, [9-12] because there are currently no fully automated methods to map laboratory data.

This thesis provides background on the history of the LOINC code system, challenges of LOINC implementation, findings from previous studies that attempted to automate laboratory mapping, and a brief overview of important concepts in machine learning. We describe the development of a novel, automated pipeline that leverages noisy labels in a machine learning algorithm to map unlabeled laboratory data and to reclassify incorrect mappings within labeled data. Using a dataset containing a mix of labeled and unlabeled data with an unknown labeling error rate, we evaluate model performance compared to manual adjudication.

Background

Primary Data Use

Clinical data use can be classified as either primary or secondary. Primary use is when clinical data is used to provide healthcare to the person from whom the data was collected. For example, when a patient's blood pressure reading is elevated (i.e. hypertension), this clinical data element may influence the healthcare provider to start an anti-hypertensive medication to lower the patient's blood pressure. Secondary clinical data use is when previously gathered information is used for purposes other than providing healthcare to the person from whom it was obtained (i.e. population health research). EHR systems are a rich source of data accumulated through routine clinical care.[4] At the point of care (primary data use), the EHR can provide comprehensive, searchable, longitudinal information about a patient, an improvement over previous paper medical records. While EHR's have dramatically changed primary data use for healthcare delivery, they also have implications on secondary data use.

Secondary Data Use and Aggregation

Secondary use of EHR data for analytics, research, quality and safety measurement, and public health is increasingly prevalent.[4,13,14] The Health Information Technology for Economic and Clinical Health (HITECH) Act[15] and the meaningful use incentive program[16] facilitated widespread EHR adoption.[17] The HITECH Act of 2009 is governmental legislation created to stimulate the adoption of EHRs and support technology in healthcare. [15] The meaningful use incentive defined standards for exchanging clinical data between healthcare providers, between healthcare providers and insurers, and between providers and patients.[16] The resulting widespread EHR adoption has made multi-site data aggregation and centralization feasible and increasingly common. These aggregate data sources are important for research, quality, public health, and commercial applications.[4] For example, analyzing aggregate data in the public health domain can facilitate early detection of emerging epidemics.[4] Additionally, such data enable the creation of generalizable observational cohorts to answer clinical questions that would not be feasible within randomized clinical trials, which are cost-prohibitive and often limited to a narrow spectrum of participants.[5] However, to harness the power of this data, one must successfully integrate the data elements collected from multiple sources. Because historical EHR implementations used locally-defined mapping, converting the data to standardized mappings is challenging, time-consuming, and resource-intensive.[8,12]

Laboratory Data

In the context of primary use, laboratory data help to diagnose and monitor disease, guide treatment, and assess patient response to treatment. For example, a glycated hemoglobin (Hemoglobin A1c) is a blood test that provides a measure of what a patient's average blood glucose levels are over weeks to months. This laboratory test can be monitored over time in patients with diabetes to modify their diabetes treatment regimen and to stratify risk for developing diabetes-related complications. In the context of secondary data use, laboratory tests are essential for developing risk-prediction models (e.g. progression of kidney disease), population-level monitoring/data mining for adverse events (e.g. liver toxicity from medications), and performing comparative effectiveness research. In both the primary and secondary data use cases, laboratory data must first be mapped to a standardized terminology so that it can be integrated, shared, and interpreted. However, this mapping process is challenging due to idiosyncratic local test naming and coding practices. The heterogeneity of laboratory test naming conventions is evident in the Veterans Affairs (VA) Corporate Data Warehouse, which contains over 320,000 distinct laboratory test names (example in **Table 1**). Because there are currently no fully automated methods to map laboratory data to their standard terminology (LOINC), mapping is resource intensive and nonscalable.

Table 1. Test Name Heterogeneity for Creatinine in VA Corporate Data Warehouse

CREAT(PRIOR TO 2/20/02)	*CREATININE mg/dL
*CREATININE	CREAT.
CREAT, MG/DL (BU)	CREATININE*IA
CREATININE, DC 1/14/16	MH CREAT, SER, mg/dl

LOINC Background

History

The LOINC common terminology for laboratory and clinical observations originated in 1994 at the Regenstrief Institute due to a growing trend of sending clinical data electronically.[6] Because laboratory tests were historically represented by local, idiosyncratic naming conventions, electronic transmissions containing locally-defined data could not be fully “understood” by the receiving system. As a result, the receiving system would either need to adopt the sender's codes, which is not feasible if receiving transmissions from multiple sources, or move toward a unified system in which the sender and receiver use the same code system. For laboratory tests, the LOINC terminology facilitates the latter solution. An example of the LOINC code fields for a serum creatinine test is shown in **Table 2**. The LOINC Component identifies the analyte, or what is being measured (e.g. hemoglobin); Property refers to the type of measurement (e.g. mass concentration, enzyme activity); Time Aspect/Timing distinguishes specimens collected at a moment in time or over a specified time interval (e.g. 24 hours); System refers to the sample from which a specimen was obtained (e.g. blood versus urine); Scale specifies the unit of measurement (e.g. quantitative or ordinal), and Method refers to the technique used to the produce the test result (e.g. automated or manual count).[18]

Table 2. LOINC Code Example: Serum Creatinine

LOINC Code	2160-0
Component	Creatinine
Property	MCnc
Timing	Pt
System	Ser/Plas
Scale	Qn
Method	*NULL*
Example Units	mg/dL
Long Name	Creatinine [Mass/volume] in Serum or Plasma
Short Name	Creat SerPl-mCnc

Abbreviations: MCnc, mass concentration; Pt, point (in time); Ser/Plas, serum or plasma; Qn, quantitative.

Challenges

Within the LOINC framework, laboratory tests can be mapped to LOINC codes that retain the same meaning across institutions. However, since LOINC contains more than 65,000 possible codes, the mapping task is non-trivial, requiring significant time and resource commitment and a good understanding of LOINC attributes for particular laboratory tests. Furthermore, even with an understanding of LOINC, human coding variation can lead to mapping inconsistencies or errors.[9,19] One study examining the mapping consistency of LOINC codes between three institutions for 100 common tests found that in 75% of cases where codes could not be matched between sites, the discrepancies were due to local coding practices.[10] Prior studies demonstrate that many of these mapping inconsistencies result from selecting codes that differ in the Property, Scale, or Methods characteristics of the codes.[9,10] Correctly translating local information to LOINC requires in-depth knowledge of laboratory testing, specifically what properties are being measured, on what entity, and by which method. However, because healthcare institutions may use vague local descriptions for tests, this information can be challenging to ascertain. For example, the Component can be difficult to determine due to idiosyncratic abbreviations (e.g. EPI-Cell, representing epithelial cell), or not specifying the type of analyte [e.g. HSV TYPE 1/2, which could represent either Herpes Simplex Virus (HSV) Type 1 or HSV Type 2]. Incomplete local information can lead to multiple mapping challenges, including no description of method (e.g. manual or automated cell count), scale (e.g. quantitative or ordinal), property (e.g. titer), timing (e.g. 24 hour) or specimen type (e.g. serum, or blood). One solution to infer some of the missing or poorly-defined aspects of institutional laboratory information is to observe test characteristics, such as frequency of testing, mean test result value, standard deviation of the value, units of measure, and/or value type (ordinal versus numeric). However, in practice, clinicians with good understanding of laboratory tests frequently do not have the fund of knowledge required to successfully translate all tests to LOINC.[10]

Prior Automated LOINC Mapping Studies

Previous studies attempting to ‘automate’ LOINC mapping through a local corpus or lexical mapping were not truly automated and still required significant manual effort.[20-22] The corpus method relies upon manually mapping local terms to LOINC codes (e.g. a local code ‘BILID’ with local description ‘Bilirubin, Direct’ that is manually mapped to LOINC code 1968-7). The lexical method attempts to map local terms to standard vocabularies such as the Unified Medical Language System (UMLS) or LOINC (e.g. ‘AST’ maps to Aspartate Transaminase in UMLS with Concept Unique Identifier [CUI] C0004002). One previously published corpus-based algorithm correctly classified the single best LOINC code 50-79% of the time within three institutions.[20] In another corpus-based algorithm including data from five Indian Health Services medical facilities, an automated tool mapped 63-76% of local laboratory tests to LOINC.[23] This study also required manual mapping of laboratory tests from all sites to LOINC codes in a ‘master file’. The automated mapping process consisted of attempting to find exact string matches between local laboratory test names and test names in the master file, which may not be generalizable outside of the Indian Health Systems. Additionally, this study did not attempt to map tests with incomplete information, which further hampers generalizability. The lexical algorithm correctly mapped 57-78% of concepts (average 63%).[21] While the generation of potential mappings in this study was automated, the method still required that an expert/clinician choose the correct mapping from a list of candidates. The Regenstrief LOINC Mapping Assistant (RELMA®) provides a semi-automated platform for mapping local terms to LOINC fields (https://loinc.org/relma). While RELMA is valuable for mapping individual site data, the user input required to execute this process when test names or units are not in a normalized format is substantial, and in our experience, increases in a non-scalable fashion when attempting to map data from multiple sites.

Scoping the LOINC Classification Problem

Because local laboratory test information contains the basic information required to map to a LOINC code, the mapping process can theoretically be automated, obviating the need for manual mapping, while improving coding consistency (by eliminating human coding variance and local coding practices. The previously defined LOINC Component, Property, Time Aspect, System, and Scale are the requisite fields that facilitate mapping laboratory data to a LOINC code. Institutional laboratory data will, at minimum, contain a local test name, specimen type, units of measurement, and the test result (numeric, categorical, or text). The local test name determines the LOINC Component; the local specimen type correlates with the LOINC System; the test name and/or specimen type may contain information pertinent to the Time Aspect (e.g. 24 hour); the local test units (e.g. mg/dL) inform the LOINC Property and Scale; and the test result itself may help determine the LOINC Scale. Because the LOINC terminology contains over 65,000 codes, defined by permutations of the LOINC fields described above, the LOINC mapping use case is an extreme multiclass classification problem in which there are complex variable interactions, a problem suited for machine learning.

Machine Learning

History and Uses

Machine learning, computational methods that use experience (i.e. learn) to improve performance and make accurate predictions, evolved from the study of pattern recognition.[24] Though the concept existed in the 1950's,[25] the field has grown tremendously over the past 20 years. This growth is in part due the mainstream availability of computing resources, improvements in computational efficiency, large data sets, and open source utilities. Machine learning algorithms have demonstrated utility in automating processes that previously required time- and resource-intensive manual work. For example, machine learning has been successfully deployed for speech recognition,[26,27] text/document classification (e.g. spam detection),[28,29] image recognition,[30] information extraction,[31-33] ranking and personalization of content,[34] and medical risk-prediction and diagnosis.[35-37]

General Definitions and Terminology

While machine learning can be used for clustering (partitioning items into similar regions), regression (predicting a real value for items), or learning associations and relations, it is most commonly used for classification (assigning categories to items).[24,38] In classification problems, machine learning scenarios can broadly be separated into supervised learning, unsupervised learning, and semi-supervised learning. *Instances* refer to data points used for learning or evaluation. *Labels* are values or categories assigned to instances. The *training sample* consists of the instances used to train a learning algorithm and the *test sample* is comprised of instances used to evaluate the learning algorithm's performance. In supervised learning, the training sample consists of labeled instances that the learner uses to make predictions on unseen data. Predicting whether or not a patient has diabetes is a case in which supervised learning can be used. In unsupervised learning, the learner uses unlabeled data to make predictions on unseen data (clustering is an example). Since no labeled instances are available in this setting, it can be difficult to quantitatively evaluate the performance of the learner/model. Finally, in semi-supervised learning, the learner uses a training sample that consists of both labeled and unlabeled data to make predictions on unseen data. This method is commonly used in settings where unlabeled data is accessible but labels are costly to obtain.

Noisy Labels

Noisy labels—the so-called 'silver standard'—have recently gained attention[39,40] because they alleviate the need to perform time-consuming manual gold standard adjudication for label assignment (i.e. forming a corpus) prior to training a classification model. Noisy labels refer to incorrect class labels resulting from an imperfect labeling process. The proportion of incorrect labels varies across domains, but commonly occurs in the 5%-40% range. For example, if a 'Creatinine, Serum' test were labeled with LOINC code 49004-5 (corresponding to Creatinine [Mass/volume] in Peritoneal dialysis fluid), that code would constitute a noisy label, because the correct code would have been 2160-0 (corresponding to Creatinine [Mass/volume] in Serum or Plasma). Implicit in noisy labeling, a large volume of training data is necessary to compensate for inaccuracy in labels (noise-tolerant learning).[41,42] Previous studies suggest large-volume, imperfectly-labeled training data can compensate for label inaccuracy and outperform models trained on smaller 'clean' datasets,[43,44] even when up to 40% of labels are incorrect.[45,46] To our knowledge, no prior studies have used noisy labels to facilitate automated mapping of laboratory tests to LOINC codes.

Multiclass Classification Methods

Mapping laboratory data to LOINC codes constitutes a multiclass classification problem. Multiclass classification is the scenario in which instances can be categorized into one of three or more classes (e.g. What Stage of breast cancer does this patient have [I, II, III, or IV]?), as opposed to binary classification, which classifies instances into one of two classes (e.g. Will this patient be readmitted to the hospital within 30 days? [yes/no]).

Common multiclass classification methods include regularized logistic regression, decision trees, neural networks, support vector machines (SVMs), and Naïve Bayes. A full discussion of these methods is beyond the scope of this project. However, it is worth noting key strengths and weaknesses of these methods: logistic regression models can be updated easily and have a probabilistic interpretation, but can perform poorly when there are multiple or nonlinear decision boundaries; neural networks reduce the need for feature engineering via hidden layers, but are computationally intensive to train and require very large amounts of data to train; SVMs can model non-linear decision boundaries and are fairly robust to overfitting, but are memory intensive and do not scale well to large datasets; and Naïve Bayes classifiers are scalable and easily to implement, but are simplistic and often perform worse than other properly tuned, trained algorithms. Classification tree methods (e.g. random forests) have gained popularity because they are robust to outliers/noise, are scalable, and naturally model non-linear decision boundaries. Individual classification trees are prone to overfitting (error when the model is too closely fit to training data), which can be alleviated by using ensemble methods that build multiple classifiers independently and average their predictions. Random forests are an ensemble learning method that constructs multiple decision trees (comprising a forest) during training, and outputs the mode of the classes. Put simply, each of the trees in the forest "votes" on the class, and the forest chooses the class having the most votes over all of the trees in the forest.

Using a dataset containing a mix of labeled and unlabeled data with an unknown labeling error rate, we hypothesized that noisy LOINC labels could be leveraged in a supervised machine learning algorithm to developed a truly automated method to map unlabeled data and reclassify incorrect mappings within labeled data. We describe our prototype model development, challenges we identified in the process, and model modification designed to address those challenges. We then present our final model and evaluate its performance when applied to both labeled and unlabeled data.

CHAPTER 2

INITIAL MODEL IMPLEMENTATION

Methods

Study Setting and Design

We collected laboratory data from the Department of Veterans Affairs (VA) Corporate Data Warehouse, which aggregates data from each VA facility's Veterans Health Information Systems and Technology Architecture (VistA) and Computerized Patient Record System (CPRS) instances.[47,48] Data included all inpatient and outpatient laboratory results from 130 VA hospitals and clinics collected between January 1, 2000 and December 31, 2016. This study was approved by the Institutional Review Board and the Research and Development Committee of the Tennessee Valley Healthcare System VA.

Data Collection and Aggregation

Within each VA site, we selected the 150 most commonly-used laboratory tests with numerically-reported results (e.g. hemoglobin, sodium). We aggregated the raw data—comprised of individual patient-level measurements—by grouping on the following data elements: 1) laboratory test identifier (a site-specific, test-specific integer), 2) specimen type identifier (a site-specific, specimen type-specific integer), 3) units of measurement, and 4) LOINC code. Within these groupings, we summarized the numeric test results using mean, median, percentiles (5th, 25th, 75th, 95th), minimum, maximum, count, and normalized frequency (the percentage of all laboratory results at the site attributed to the specific test). Each data row formed by aggregation comprised an instance.

Feature Engineering

Automated Text Processing

We processed source data test name and specimen type first creating a dictionary of unique test names and specimen types by site. We then removed punctuation, dates, and stop words from all test names and specimen types (**Table 3**).

Table 3. Source Data Processing

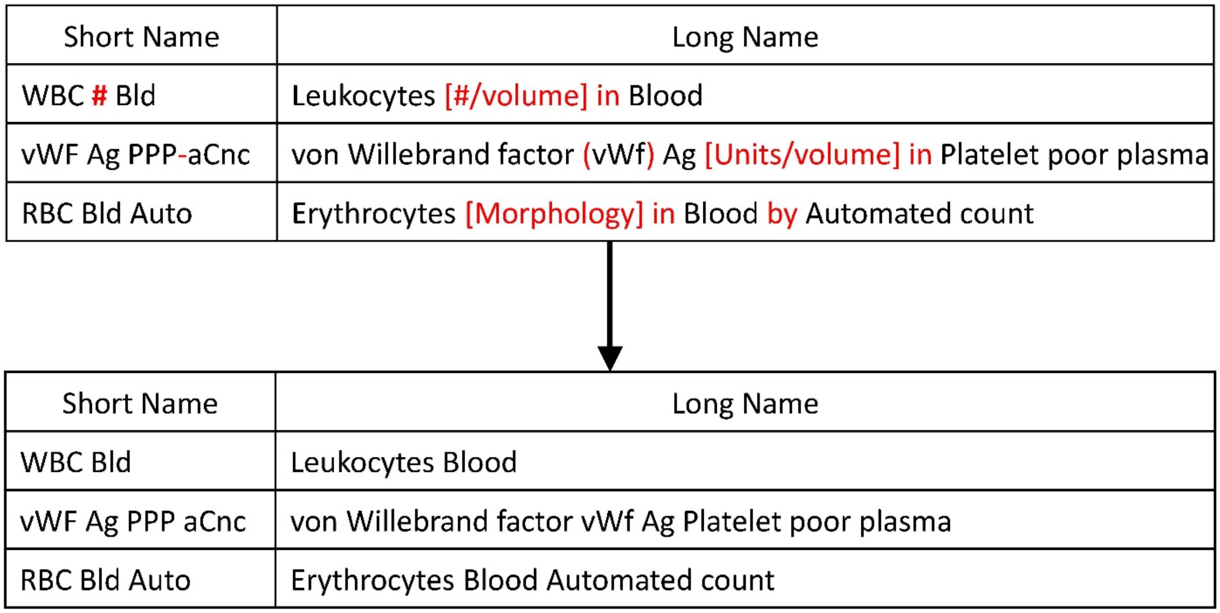
Site	Original Test Name	Processed Test Name
1	B12 (9/14/11)	B12
2	Gamma Globulin (EP)	GAMMA GLOBULIN EP
3	HGB A1C (6/8/98-3/16/09)	HGB A1C
3	.CK MB ISO (BY ECI) (TO 4/14/08)	CK MB ISO ECI

LOINC Table Data Preprocessing

We used the publicly-available LOINC® table (version 2.56), restricting to the laboratory and clinical observation class types,[6] in automated feature generation. We preprocessed the Short Name and Long Name fields by removing punctuation, stop words, and bracketed phrases (**Figure 1**). We then computed the co-occurrence between each LOINC Short Name token and a sliding window of 1 to 3 LOINC Long Name tokens (**Figure 2**). We used a window length upper bound of 3 because the long-form components of most medical acronyms are ≤ 3 words. From this output, we mapped each LOINC Short Name token to the Long Name token or phrase with the highest count by co-occurrence. In the case of ties, we evaluated whether the group of Long Name phrases with the highest co-occurrence contained an expansion of a Short Name token acronym (i.e. each letter of the Short Name token corresponded to the first letter of Long Name tokens). If no acronym expansion was detected within a group, we mapped the

Short Name token to the shortest phrase in the Long Name group (**Figure 3**). This process resulted in a cross-walk from LOINC Short Name Tokens to LOINC Long Name tokens/phrases.

Figure 1. LOINC Table Data Preprocessing

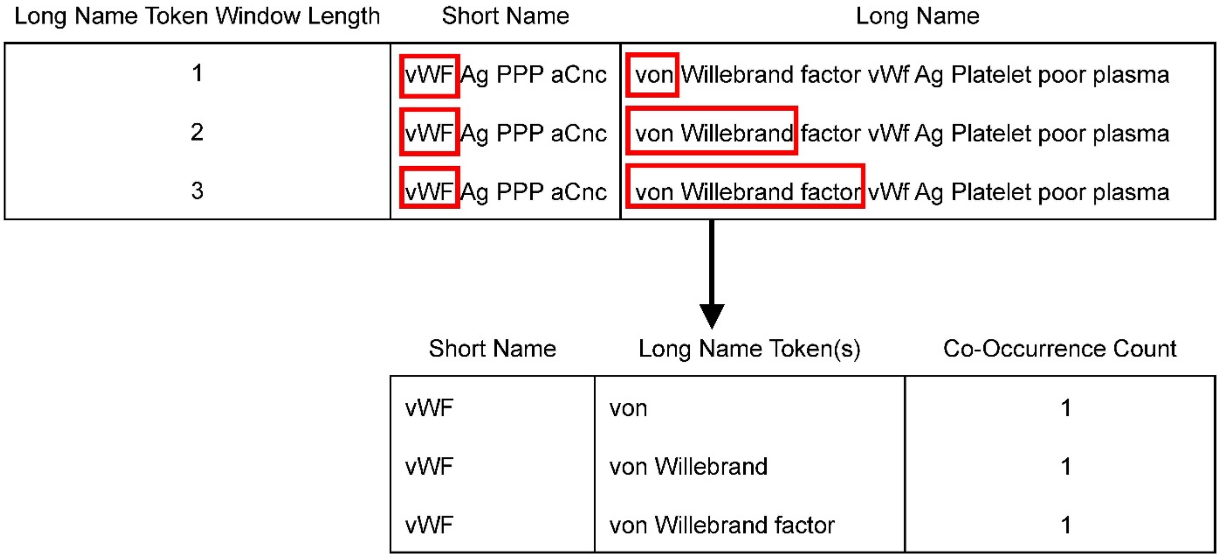


Short Name	Long Name
WBC # Bld	Leukocytes [#/volume] in Blood
vWF Ag PPP-aCnc	von Willebrand factor (vWf) Ag [Units/volume] in Platelet poor plasma
RBC Bld Auto	Erythrocytes [Morphology] in Blood by Automated count

Short Name	Long Name
WBC Bld	Leukocytes Blood
vWF Ag PPP aCnc	von Willebrand factor vWf Ag Platelet poor plasma
RBC Bld Auto	Erythrocytes Blood Automated count

The upper table represents fields in the unprocessed publicly-available LOINC table. Punctuation, stop words, and bracketed phrases (highlighted in red in the top table) were removed. The bottom table represents the LOINC Short Name and Long Name fields after

Figure 2. Computing LOINC Short Name and Long Name Token Co-occurrence using a 3-token Window



Long Name Token Window Length	Short Name	Long Name
1	vWF Ag PPP aCnc	von Willebrand factor vWf Ag Platelet poor plasma
2	vWF Ag PPP aCnc	von Willebrand factor vWf Ag Platelet poor plasma
3	vWF Ag PPP aCnc	von Willebrand factor vWf Ag Platelet poor plasma

Short Name	Long Name Token(s)	Co-Occurrence Count
vWF	von	1
vWF	von Willebrand	1
vWF	von Willebrand factor	1

Figure 3. Handling Ties in LOINC Term Co-occurrence Counts and Selecting Acronyms

Short Name Token	Mapping	Count
RBC	Red	383
RBC	Red Blood	383
RBC	Red Blood Cells	383
WBC	Leukocytes	137
vWF	von	15
vWF	von Willebrand	15
vWF	von Willebrand Factor	15

Short Name Token	Mapping	Count
RBC	Red Blood Cells	383
WBC	Leukocytes	137
vWF	von Willebrand Factor	15

Each LOINC Short Name token was mapped to the Long Name token or phrase with the highest count by co-occurrence. In the case of ties, we first determined if the group of Long Name phrases with the highest co-occurrence contained an expansion of a Short Name token acronym. If no acronym expansion was detected within a group, the Short Name token was mapped to the shortest phrase in the Long Name group. In the above example, for the Short Name Token 'RBC' the highest co-occurrences contained ties. Each letter of the Short Name token corresponds to the first letter of each of the Long Name tokens, thus 'RBC' is mapped to 'Red Blood Cells.' In the case of the 'WBC' token, the highest co-occurrence count contains no ties, and is simply mapped to 'Leukocytes'.

To handle abbreviations contained in the LOINC System field, we used string distance-matching with the Jaro-Winkler metric[49-51] to find the corresponding words with the smallest edit distance in the LOINC Long Name field. We mapped the System token to the resulting distance-matched Long Name token, except when the System could be mapped to an acronym expansion (in which case the acronym expansion was used).

LOINC Feature Engineering

We string distance-matched tokens from the processed source data test names and specimen types to the tokens derived from the LOINC data preprocessing step described above. We used the Jaro-Winkler[49-51] and Levenshtein[52] distance metrics for string distance-matching. The Jaro-Winkler algorithm measures the number of characters that two strings have in common, taking into account matches and transpositions. Because differences between two strings may be more important if they occur at the start of the string, the Jaro-Winkler method includes a correction factor that more favorably rates string pairs that match at the beginning rather than the end. The Levenshtein algorithm provides a count of the number of insertions, deletions, or substitutions required to convert one string to the other. We included both metrics for feature generation because they often provide different results. Using these string distance metrics, for each test name and specimen type we identified the LOINC Long Common Name token with the smallest edit distance. We concatenated the resulting tokens from the Long Name to form the two ‘LOINC Long Name mapped from Test Name’ features (one text string for each distance-matching metric) and the two ‘LOINC Long Name mapped from Specimen Type’ features. We distance-matched the resulting mapped test names and specimen types, using both the Jaro-Winkler and Levenshtein metrics, to the LOINC Component and System fields, respectively. We included the predicted Component and System, along with their corresponding match distances from each of the two string distance-matching metrics, as model features. The features included in the initial model are shown in **Table 4**.

Table 4. Initial Model Features

Text	Numeric
LOINC Long Name (JW) mapped from Test Name	Test result 5th percentile
LOINC Long Name (LV) mapped from Test Name	Test result 25th percentile
LOINC System (JW) mapped from Specimen Type	Test result median
LOINC System (LV) mapped from Specimen Type	Test result mean
Predicted LOINC Component (JW)	Test result 75th percentile
Component Match Distance (JW)	Test result 95th percentile
Predicted LOINC Component (LV)	Test result minimum
Component Match Distance (LV)	Test result maximum
Predicted LOINC System (JW)	Normalized test frequency*
System Match Distance (JW)	
Predicted LOINC System (LV)	
System Match Distance (LV)	
Units	

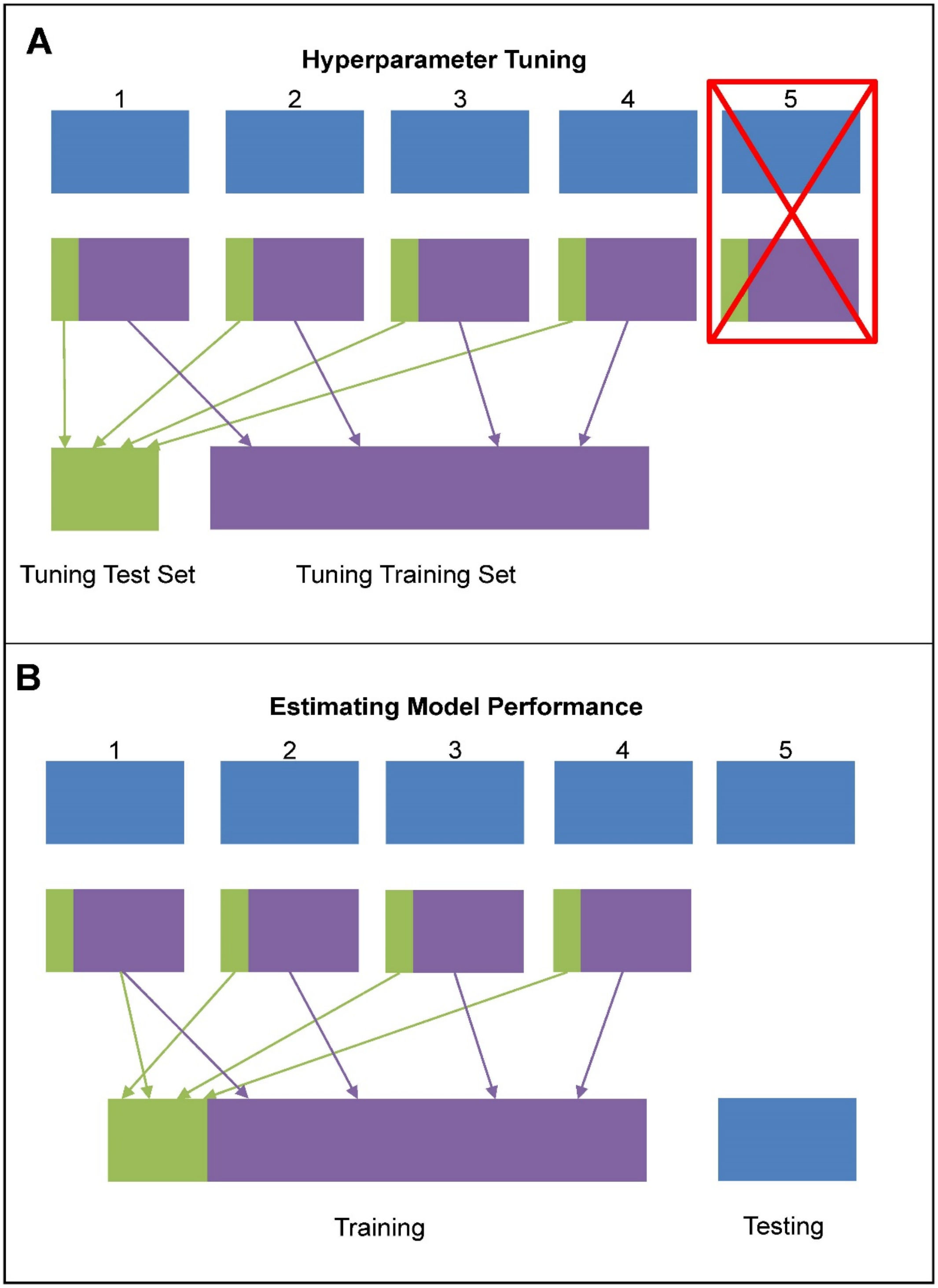
JW, Jaro-Winkler; LV, Levenshtein.

* Normalized test frequency calculation $= \frac{\text{Test frequency}}{\text{Total number of laboratory results per site}} \boldsymbol{X}\ \mathbf{100}\%$

Data Filtering and Partitioning

We held out data instances with missing specimen type and/or LOINC code in the unlabeled dataset for a separate analysis. Additionally, instances containing LOINC codes with only one occurrence by test volume or comprising fewer than 5 instances in the aggregate dataset were combined with the unlabeled data for reclassification, under the assumption that these labels might be incorrect. In the remaining labeled dataset, we partitioned data for 5-fold cross-validation using splits by sites (**Figure 4**).

Figure 4. Data Partitioning for Hyperparameter Tuning and Estimating Model Performance



Within each of the 5-fold split-by-site data partitions (blue boxes), we performed 80/20 splits for hyperparameter tuning. **A)** Example of the first of five cross-validation iterations for hyperparameter tuning. Within the first 4 data blocks, the tuning training set (purple) is used to fit the model with different hyperparameters and the tuning test set (green) is used to estimate error. For hyperparameter tuning, the 5th data partition is not used. **B)** Example of one of five cross-validation iterations for estimating model performance. In this example, all data from the first 4 blocks is used for model training (green and purple), and the 5th block (blue) is used for testing.

Machine Learning Models

We implemented two machine learning models, a random forest multiclass classifier, and a one-versus-rest ensemble of binary random forest classifiers. Model building and analyses were conducted using scikit-learn in Python.[53] Using 5-fold cross-validation as depicted in **Figure 4A** and accuracy (number of correct predictions / total number of predictions) as the loss function, hyperparameters were manually tuned in step-wise fashion in the following order: 1) maximum features per split, 2) maximum tree depth, 3) minimum samples per split, and 4) maximum number of estimators.

Estimating Model Performance

Using 5-fold cross-validation with the aforementioned splits by site within the labeled dataset, we estimated model performance for the random forest and one-versus-rest models. For our initial model, we used only the accuracy measure due to its intuitive interpretation.

Model Fitting and Label Assignment

We fit the random forest and one-versus-rest models using the entire labeled dataset. We then used the fitted models to predict LOINC codes on the holdout dataset comprised of instances with either missing or infrequently-used LOINC codes.

Manual Validation

We used the label predicted in model with best cross-validated performance for manual validation. Initial manual validation was conducted by a single physician reviewer. In the unlabeled data, we randomly sampled 200 instances. From the labeled dataset, we randomly selected 100 instances where the predicted LOINC code matched the original LOINC code (concordant), and 100 instances in which the predicted LOINC code did not match the original LOINC code (discordant).

Results

The raw laboratory data was comprised of approximately 6.6 billion test results, ranging from 2.5 to 184 million results per site. After aggregating by laboratory test identifier, specimen type identifier, units, and LOINC code, the analytic dataset consisted of 140,565 instances and 2,215 distinct LOINC codes. LOINC codes were missing in 44,199 instances of source data, comprising to a total test volume of approximately 450 million results. This corresponds to missing LOINC codes in 31% of the data by instance, and 7% of the data by test frequency.

Model Performance on Labeled Data

For both the random forest multiclass classifier and the one-versus-rest ensemble of binary random forest classifiers, we evaluated performance when features were restricted to text elements, numeric features, and including all features (**Table 5**). Using both text and numeric features, the multiclass and one-versus-rest models performed comparably, with accuracy of 57-58%. In both models, performance was driven by the text features, which when used alone for model training yielded accuracy of approximately 54%. Numeric features alone resulted in poor model performance, with accuracy of about 33%, but when combined with the text features provided incremental improvement in accuracy.

We examined the top 15 feature importances for the initial random forest classifier and the one-versus-rest ensemble classifier. In the random forest model, text features clearly outperformed numeric features (**Figure 5**), while the one-versus-rest model had lower individual feature importances, with numeric and text features both adding predictive value (**Figure 6**).

Table 5. Initial Model Predictive Accuracy Stratified by Feature Type

		Accuracy
Random Forest	Numeric Features	33.7%
	Text Features	54.3%
	All Features	56.8%
One-Versus-Rest	Numeric Features	33.3%
	Text Features	53.6%
	All Features	57.7%

Figure 5. Feature Importance in Initial Multiclass Random Forest Model

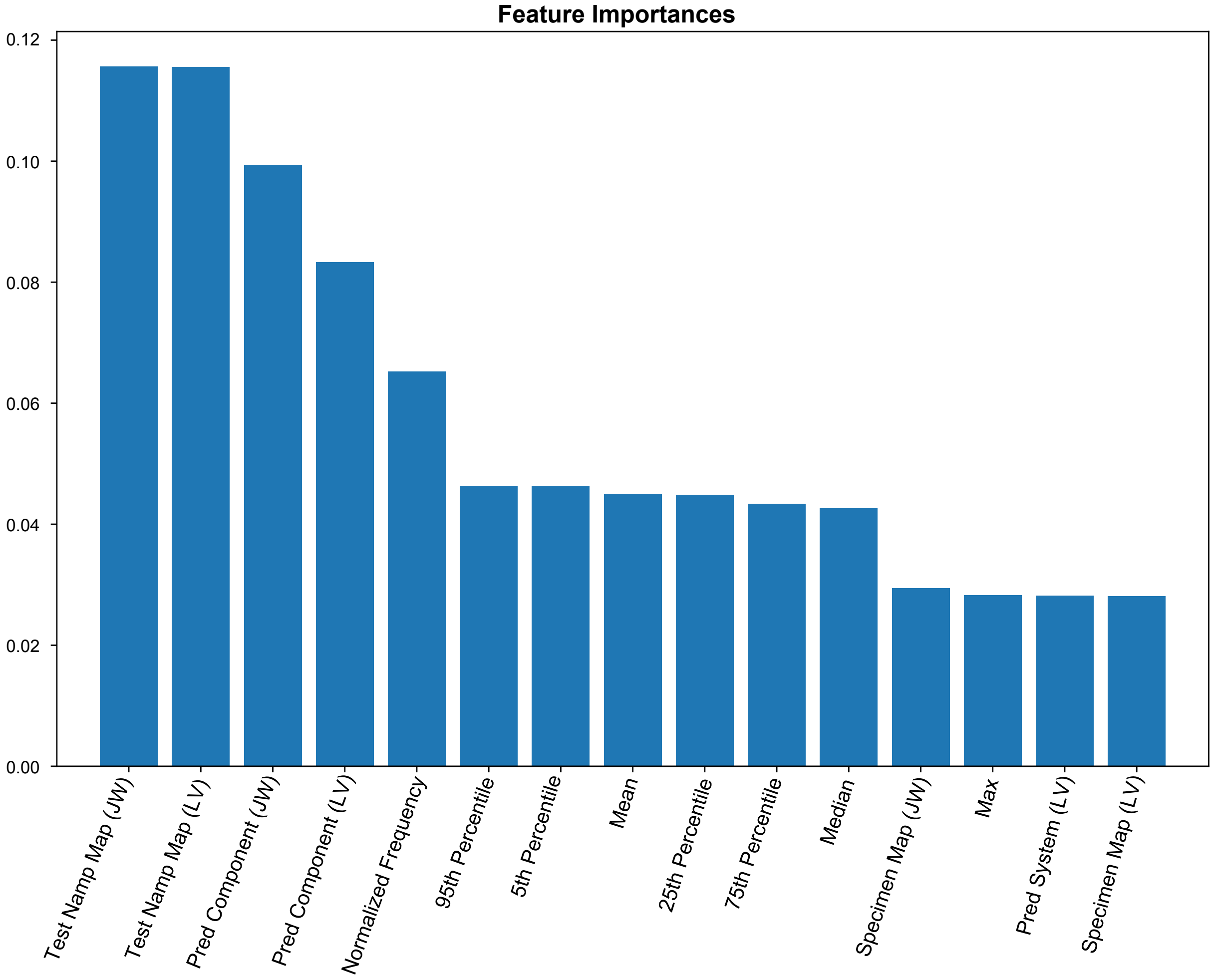
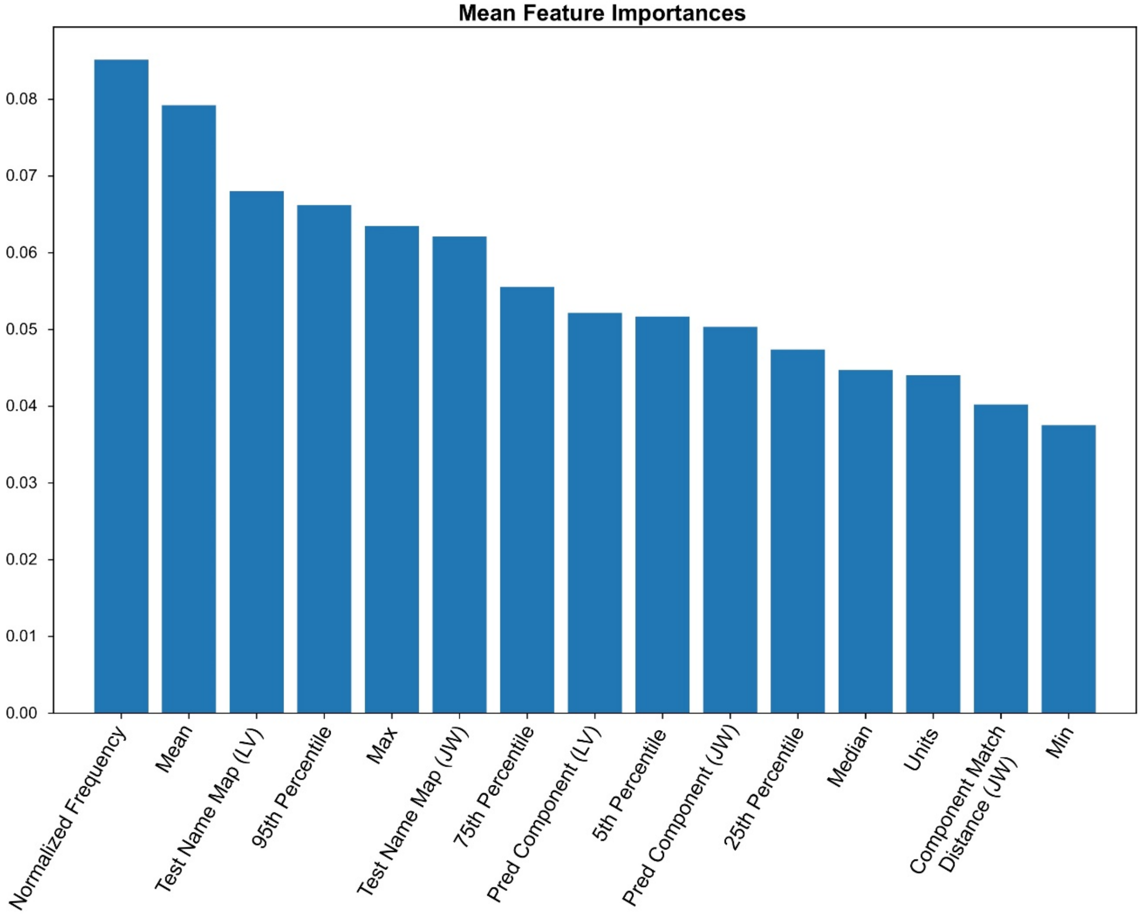


Figure 6. Mean Feature Importance in Initial One-versus-rest Ensemble of Binary Random Forest Classifiers



Manual Validation

Unlabeled Data

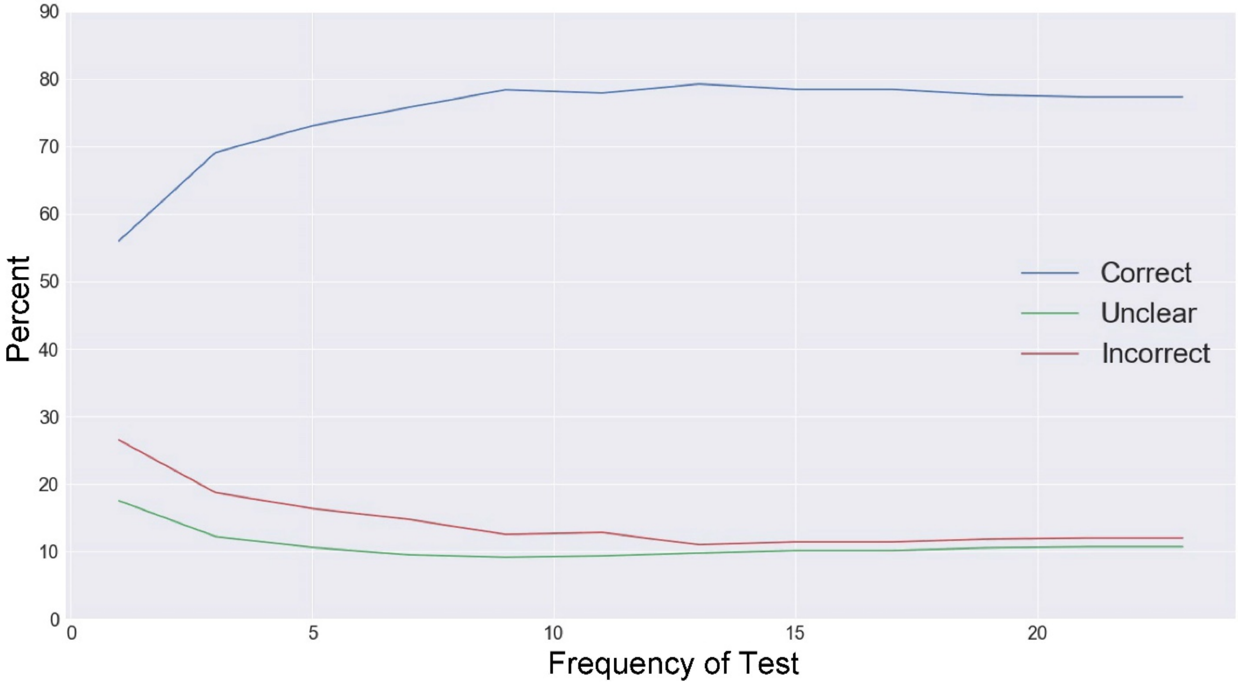
In initial manual validation of the unlabeled dataset using the model-predicted label from the one-versus-rest classifier, the model-predicted label was correct in 91% of sampled instances, and 92% of tests (**Table 6**). The model-predicted label accuracy was maximal when the test frequency was at least 10 (**Figure 7**).

Table 6. Initial Manual Evaluation of LOINC Codes Assigned to Unlabeled Data

Predicted LOINC Code	Instances (N=200)	Tests (N=3,364,245)
Correct	182 (91%)	3,097,999 (92.1%)
Incorrect	10 (5%)	263,744 (7.8%)
Insufficient Information to Determine	8 (4%)	2,502 (0.1%)

Definition: Insufficient Information to Determine: Either not enough source data to infer code (i.e. units missing and would be necessary to assign code), or source data conflicts (i.e. test name includes the word 'blood' and specimen type is 'urine').

Figure 7. Initial Model Performance in Unlabeled Dataset Based on Test Frequency



Labeled Data

In the labeled data, when the original source data LOINC code and the model-predicted LOINC code were equal (concordant), the labels were correct in 92.0% of sampled instances and 99.9% of associated tests (top of **Table 7**). In cases where the original source data LOINC code and the model-predicted LOINC code were not equal (discordant), the model-predicted LOINC code was correct in 68.0% of instances and 94.2% of associated tests, and the model-predicted code was better than the original code in 45.0% of instances (16.6% of associated tests). In 23% of instances (77.6% of associated tests), both the original and model-predicted LOINC codes were correct due to LOINC code equivalence.

Table 7. Initial Manual Evaluation of LOINC Codes Assigned to Labeled Data

Concordant Original and Predicted LOINC	Aggregate Data Instances (N=100)	Associated Tests (N=14,910,333)
Correct	92	14,895,799 (99.9%)
Incorrect	2	316 (<0.1%)
Insufficient Information to Determine	6	14,218 (0.1%)

Discordant Original and Predicted LOINC	Aggregate Data Instances (N=100)	Associated Tests (N=1,515,472)
Total Correct	68	1.427,968 (94.2%)
Predicted LOINC Correct	45	251,498 (16.6%)
Both Correct (Equivalent LOINC Codes)	23	1,176,470 (77.6%)
Total Incorrect	28	86,918 (5.7%)
Original LOINC Correct	19	85,113 (5.6%)
Both Incorrect	9	1,805 (0.1%)
Insufficient Information to Determine	4	585 (<0.1%)

Definitions: Both Correct (Equivalent LOINC Codes): Model-predicted label ≠ original label and both labels are correct due to LOINC synonymy; Insufficient Information to Determine: Either not enough source data to infer code (i.e. units missing and would be necessary to assign code), or source data conflicts (i.e. test name includes the word ‘blood’ and specimen type is ‘urine’).

Challenges

Uninformative Tokens in Test Names

Discovery

Upon manually reviewing the output of cleaned source data test names through the automated text processing module, we discovered that after the cleaning step some tokens remained that did not contribute meaning to the test name. For example, terms such as ‘sendout’, and ‘DC’d’ are not informative for determining what a particular test signifies.

Method Modification

For each token, defined as a string of one or more alphanumeric characters separated by white space, we added a function to the text processing module to calculate the percent occurrence of each token as a function of the total number of tokens per site. Using a tunable threshold (based on manual inspection), tokens occurring above a certain frequency within a site can be removed and recorded in a separate comma-separated values file for review.

Handling LOINC Table Synonyms

Discovery

During manual review of the mappings from source data test names to LOINC long names (completed using string distance matching), we confirmed that by using the publicly available LOINC table, we were not able to handle term synonymy. For example, a source data test name of ‘Dilantin’, maps to a LOINC Long Name token of ‘cilantro’ (Levenshtein metric), or ‘plantain’ (Jaro-Winkler metric). The LOINC table does not contain the term Dilantin, and instead includes phenytoin, the generic name for Dilantin.

Method Modification

To handle term synonymy, we initially used the Apache clinical Text Analysis and Kowledge Extraction System (cTAKES).[54] Due to cumbersome implementation, requirement for large heaps of random access memory (RAM), inefficient computing time, and redundancy in output, cTAKES was abandoned in favor of the Unified Medical Language System (UMLS) Representational State Transfer (REST) Application Program Interface (API). With the processed text from the aforementioned output, we used the UMLS REST API to obtain UMLS concept unique identifiers (CUIs) for test names and specimen types, respectively. Adding this process to our algorithm allowed us to handle synonyms not captured by string distance-matching using the LOINC table alone. Furthermore, by incorporating UMLS CUIs, we were able to generate more features for the machine learning algorithms.

LOINC Equivalence

Discovery

From the manual validation of initial model performance on labeled data (**Table 7**), we discovered that in 78% of the cases where the predicted LOINC code differed from the original LOINC code, the codes were actually equivalent. In a focused literature review, we found previously published work reporting that there are three levels of interoperability that may exist between two LOINC codes.[55] In Level I interoperability, the LOINC Component, Time Aspect, Scale, Property, System, and Method are identical for two codes. In Level III interoperability, the two codes differ only the LOINC Method (example in **Table 8**). In the latter scenario, two codes can be used interoperably (albeit, with some meaning loss) in cases where the method is not considered important. In this study we did not consider Level II interoperability, which requires data processing to make codes comparable (e.g. log conversion).

Table 8. Example of Level III LOINC Code Interoperability

LOINC	Component	Property	Timing	System	Scale	Method
26471-3	Leukocytes other/100 leukocytes	NFr	Pt	Bld	Qn	
40646-2	Leukocytes other/100 leukocytes	NFr	Pt	Bld	Qn	Automated count
730-2	Leukocytes other/100 leukocytes	NFr	Pt	Bld	Qn	Manual count

We also reviewed the new publicly-available LOINC Groups table.[56] However, we did not use this newly released publication, which contains multiple different equivalence types in the same table, and does not presently include a way for users to distinguish equivalence type. Additionally this publication has not yet been validated.

Method Modification

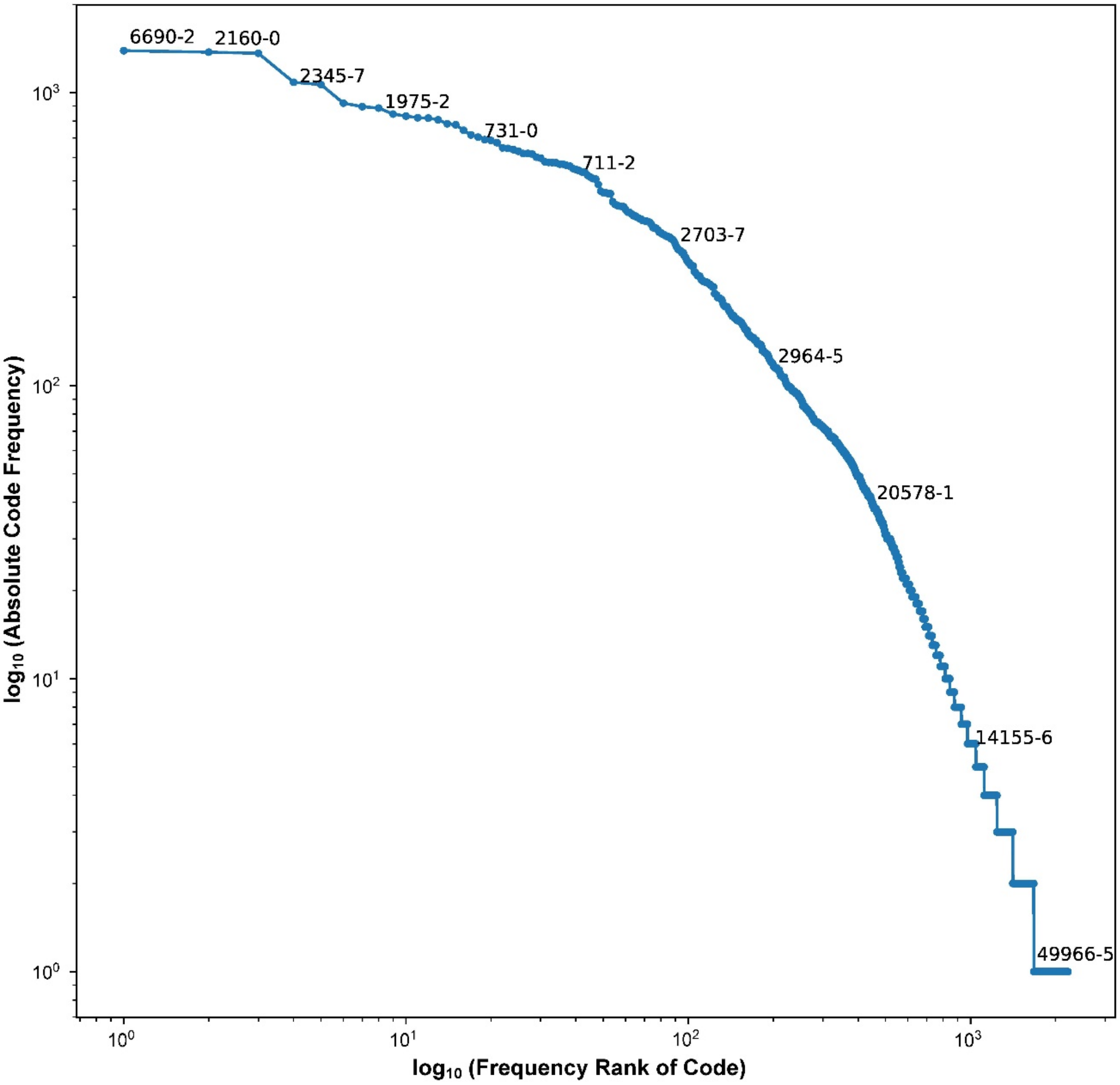
To automate the creation of equivalent LOINC code groups, we first grouped LOINC codes by Component, Property, Time Aspect, System, and Scale. Within these groups, we created key-value pairs for groups with Level I (identical methods) or Level III interoperability (differing methods). For Level I interoperable groups with a specified method and only one LOINC code with status of ‘Active’, this code was defined as the key, with all other codes in the group forming the values. For Level III interoperability in groups containing only one methodless LOINC code, the methodless code was defined as the key with all other LOINC codes in the group comprising the values. For groups containing more than one methodless code and only one code with ‘Active’ status, this code was defined as the key. In groups containing more than one methodless code and no codes with ‘Active’ status, but a single code with ‘Discouraged’ status, then this code formed the key for the group. Finally, if all codes within a group had status of ‘Deprecated’, the key was defined as the methodless code, except in cases where multiple methodless codes existed for the group, in which case the last LOINC code was selected.

Multi-class Classification and Class Imbalance

Discovery

The source data contained 2,215 distinct LOINC codes, with the most common code occurring 1,391 times in the dataset (top left, **Figure 8**), and the least common codes occurring only once (bottom right, **Figure 8**). We examined the number of labels in the test set that were

Figure 8. LOINC Code Frequency Rank versus Absolute Code Frequency



not present in the training set for the cross-validation steps in both the hyperparameter tuning. In the test set for model hyperparameter tuning, on average 14 (1.6%) of the labels were not present in the training dataset (**Table 9**), compared with 23 (2.3%) in cross-validation using the full dataset (**Table 10**). In both scenarios, the labels not present in the training dataset were associated with a very small number of tests (<0.1%).

Table 9. Label Imbalance in Initial Model Hyperparameter Tuning

Distinct Test Set Labels	Distinct Training Set Labels	Test Set Labels not Present in Training Set	Number of Tests Associated with Unbalanced Test Set Label	Total Tests in Test Set
1,035	1,435	19 (1.84%)	61,589 (0.01%)	775,504,186
1,042	1,426	17 (1.63%)	133,672 (0.02%)	750,933,489
1,042	1,437	16 (1.54%)	58,202 (0.01%)	837,147,736
1,062	1,457	14 (1.32%)	338,506 (0.04%)	793,678,903
1,039	1,440	19 (1.83%)	276,341 (0.03%)	801,861,074

Table 10. Label Imbalance in Initial Model Cross-validation Training

Distinct Test Set Labels	Distinct Training Set Labels	Test Set Labels not Present in Training Set N(%)	Number of Tests Associated with Unbalanced Test Set Label N(%)	Total Tests in Test Set
1,092	1,454	27 (2.47%)	796,190 (0.06%)	1,387,362,728
1,095	1,443	38 (3.47%)	1,340,312 (0.10%)	1,404,256,060
1,070	1,453	28 (2.62%)	699,480 (0.07%)	974,756,279
1,009	1,471	10 (0.99%)	20,857 (<0.01%)	1,165,458,633
1,078	1,459	22 (2.04%)	211,638 (0.02%)	1,174,925,542

Method Modification

To reduce the number of classes in this extreme multi-class classification problem, we first implemented the automated roll-up of LOINC codes into LOINC group keys as described above. After transforming LOINC codes to their respective LOINC keys, we were left with 1,895 distinct LOINC keys in the unfiltered dataset. Instances containing LOINC keys that occurred at only one site or <10 times by test volume were combined with the unlabeled data for reclassification, because their rarity suggested that the codes present in the source data may be suspect. Following LOINC roll-up, the average number of labels in the test sets for hyperparameter tuning and cross-validation were reduced to 861 and 884, respectively, with concomitant decreases in the number and percentage of labels not present in the test sets for tuning (N=9 [1.1%]) and cross-validation for model performance estimates (N=6 [0.7%]).

To handle class imbalance, we considered oversampling. However, oversampling is a valid approach only when it can be performed *after* splitting data into training and testing sets, otherwise performance will be optimistic (i.e. the model appears to perform better than it will generalize, because training data is ‘bled’ into the testing dataset). We also considered implementing a data-driven approach to split/filter data in a way that would ensure that all labels present in the testing set were also represented in the training set. However, because this method may not be generalizable to future use cases, we opted to build a model that can handle class imbalance independent of the complexities of the specific data source.

Because some LOINC codes are more commonly used than others, class imbalance is observed in this dataset and is likely to be an issue in future use cases. In our initial model, for simplicity we used the accuracy measure to tune model hyperparameters and also to examine model performance. However, with class imbalance, accuracy can be driven by simply predicting the labels of the most common classes. Thus, for the final model, we used the weighted F1 score to tune hyperparameters. F1 score is the harmonic mean between precision and recall, and weighting refers to the support, or number of true instances per label. For evaluating model performance, in addition accuracy, we report the weighted F1 score and the micro-averaged F1 score.

CHAPTER 3

FINAL MODEL

Methods

Study Setting and Design

As presented in the initial model development, we collected laboratory data from the Department of VA Corporate Data Warehouse.[47,48] Data included all inpatient and outpatient laboratory results from 130 VA hospitals and clinics collected between January 1, 2000 and December 31, 2016. The study was approved by the Institutional Review Board and the Research and Development Committee of the Tennessee Valley Healthcare System VA.

Data Collection and Aggregation

Unchanged from the initial model development, we selected the 150 most common laboratory tests with numerically reported results for each site. We aggregated the raw data—comprised of individual patient-level measurements—by grouping on the following data elements: 1) laboratory test identifier (a site-specific, test-specific integer), 2) specimen type identifier (a site-specific, specimen type-specific integer), 3) units of measurement, and 4) LOINC code. Within these groupings, we summarized the numeric test results using mean, median, percentiles (5th, 25th, 75th, 95th), minimum, maximum, count, and normalized frequency (the percentage of all laboratory results at the site attributed to the specific test). Each data row formed by aggregation comprised an instance (example shown in **Table 11**).

Table 11. Example of Instances Created by Data Aggregation

Site	Lab Test Name	**Lab Test ID**	Specimen	**Specimen ID**	**Units**	**LOINC**	Min	Max	5th Percentile	25th Percentile	Median	75th Percentile	95th Percentile	Count	Normalized Frequency
1	CREATININE	120006	PLASMA	120000	mg/dL	*Missing*	0.1	40	0.6	0.9	1.0	1.2	2.1	85,110	0.04623
1	CREATININE	120006	PLASMA	120000	Null	2160-0	0.5	6	0.7	0.9	1.1	1.2	2.4	48	0.00003
1	CREATININE	120006	PLASMA	120000	mg/dL	2160-0	0.1	103	0.7	0.8	1.0	1.2	2.1	4,365,542	2.37134
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
2	ALBUMIN	100004	SERUM	100000	G/dL	*Missing*	0.3	5.8	3.0	3.9	4.2	4.4	4.7	55,779	0.03283
2	ALBUMIN	100004	SERUM	100000	G/dL	1751-7	0.5	9.9	3.0	3.9	4.2	4.4	4.7	941,757	0.54675
2	ALBUMIN	100004	PLASMA	100000	G/dL	*Missing*	0.4	5.5	2.8	3.7	4.0	4.2	4.5	36,077	0.02095
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

Data aggregated by grouping on the following four data elements: 1) laboratory test identifier (a site-specific, test-specific integer), 2) specimen type identifier (a site-specific, specimen type-specific integer), 3) units of measurement, and 4) LOINC code. Within these groupings, we summarized the numeric test results using mean, median, percentiles (5th, 25th, 75th, 95th), minimum, maximum, count, and normalized frequency.

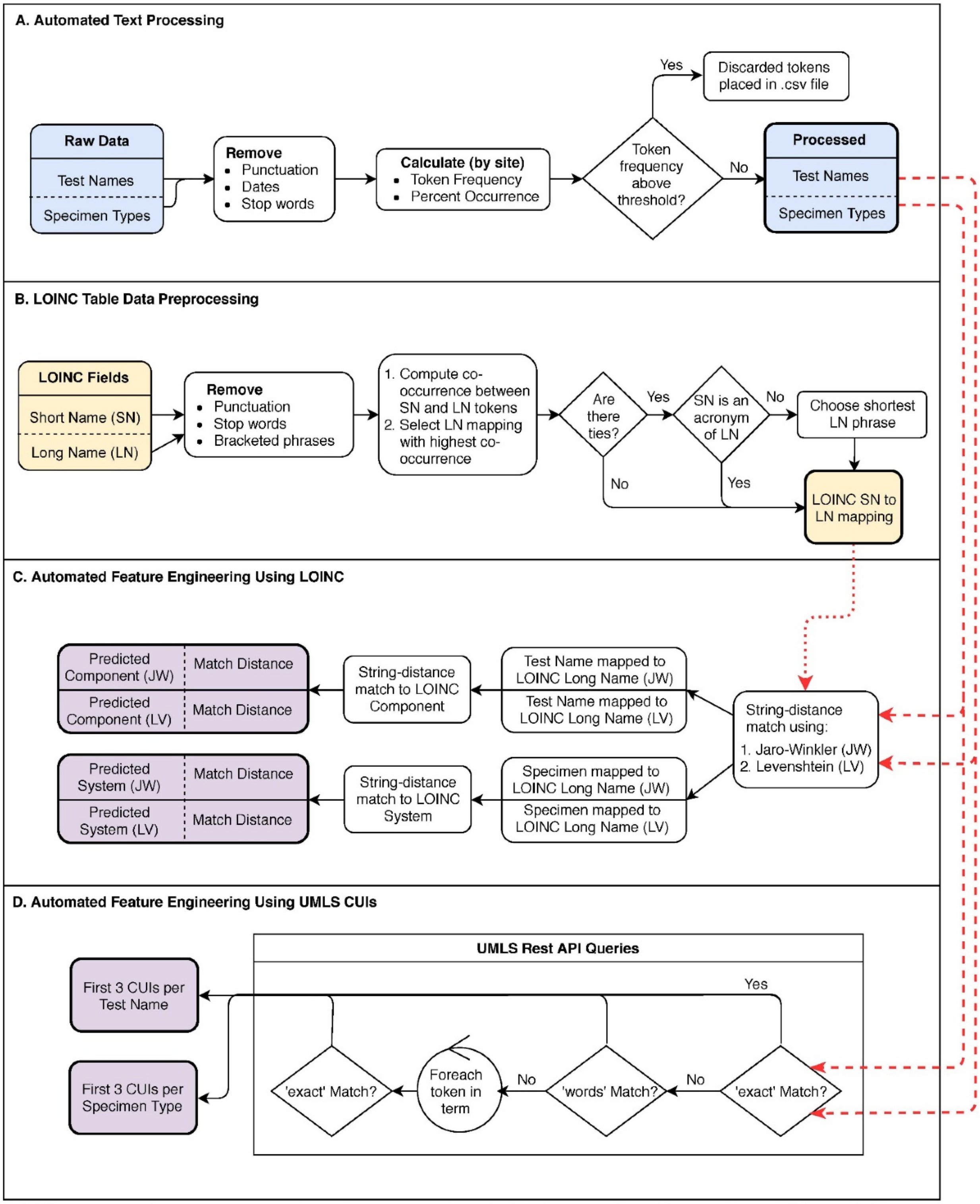
Ancillary Data Sources

We used the publicly-available LOINC® table (version 2.56) for automated feature generation, restricting to the laboratory and clinical observation class types.[6] We also used the Unified Medical Language System (UMLS®) REST API to generate model features containing Concept Unique Identifiers (CUIs).[57]

Feature Engineering

The schematic of data processing and feature engineering is depicted in **Figure 9A-D** and described further in the text that follows.

Figure 9. Data Processing and Feature Engineering


A. Automated Text Processing
Raw Data
Test Names
Specimen Types
Remove
• Punctuation
• Dates
• Stop words
Calculate (by site)
• Token Frequency
• Percent Occurrence
Token frequency above threshold?
Yes
Discarded tokens placed in .csv file
No
Processed
Test Names
Specimen Types
B. LOINC Table Data Preprocessing
LOINC Fields
Short Name (SN)
Long Name (LN)
Remove
• Punctuation
• Stop words
• Bracketed phrases
1. Compute co-occurrence between SN and LN tokens
2. Select LN mapping with highest co-occurrence
Are there ties?
Yes
No
SN is an acronym of LN
No
Yes
Choose shortest LN phrase
LOINC SN to LN mapping
C. Automated Feature Engineering Using LOINC
Predicted Component (JW)
Match Distance
Predicted Component (LV)
Match Distance
String-distance match to LOINC Component
Test Name mapped to LOINC Long Name (JW)
Test Name mapped to LOINC Long Name (LV)
String-distance match using:
1. Jaro-Winkler (JW)
2. Levenshtein (LV)
Predicted System (JW)
Match Distance
Predicted System (LV)
Match Distance
String-distance match to LOINC System
Specimen mapped to LOINC Long Name (JW)
Specimen mapped to LOINC Long Name (LV)
D. Automated Feature Engineering Using UMLS CUIs
UMLS Rest API Queries
First 3 CUIs per Test Name
First 3 CUIs per Specimen Type
Yes
'exact' Match?
Foreach token in term
No
'words' Match?
No
'exact' Match?

A) Raw source data test name and specimen type automated text processing. **B)** From the publicly-available LOINC table, processing of LOINC Short Name (SN) and Long Name (LN) fields and mapping of SN tokens to LN tokens/phrases, **C)** String distance-matching tokens from the processed source data test names and specimen types (from A) to the tokens derived from the LOINC data preprocessing step (from B), with final mapping to the predicted LOINC Component and System fields. **D)** Using the UMLS REST API to obtain UMLS CUIs for test names and specimen types (from A).

Automated Text Processing

We processed source data test name and specimen type by first removing punctuation, dates, and stop words (**Figure 9A**). For each token, we computed the percent occurrence as a function of the total number of tokens per site. Using a tunable threshold (4% in this study based on manual inspection of the output), tokens occurring above a certain frequency within a site were removed (example in **Table 12**).

Table 12. Example of High Frequency Terms Removed from Source Data

Site	Discarded Term
358	SENDOUT
402	DCD
459	OLD
516	THRU
516	DCD
521	CVICU

LOINC Table Data Preprocessing

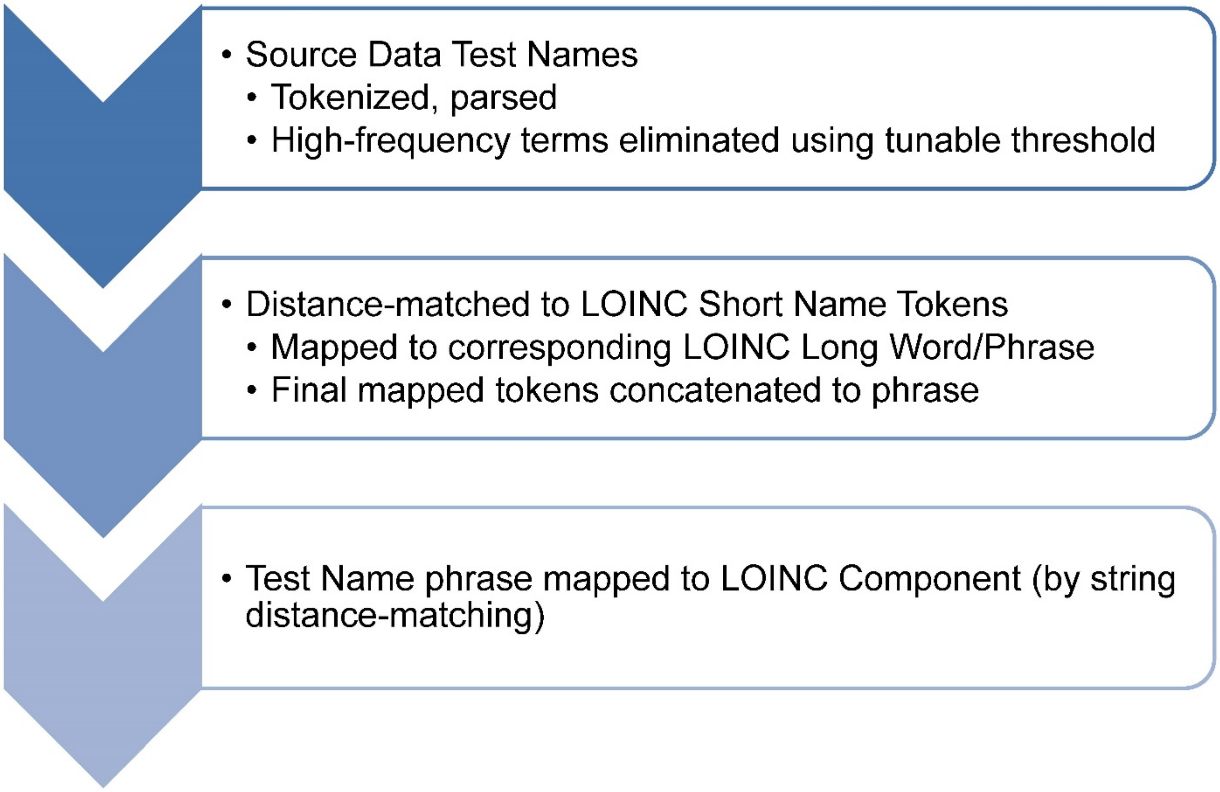
For the final model, we made no changes to the LOINC table preprocessing algorithm presented in the initial model. From the publicly-available LOINC table, we preprocessed the Short Name and Long Name fields by removing punctuation, stop words, and bracketed phrases (**Figure 9B**). We then computed the co-occurrence between each LOINC Short Name token and a sliding window of 1 to 3 LOINC Long Name tokens. From this output, we mapped each LOINC Short Name token to the Long Name token or phrase with the highest count by co-occurrence. In the case of ties, we evaluated whether the group of Long Name phrases with the highest co-occurrence contained an expansion of a Short Name token acronym (i.e. each letter of the Short Name token corresponded to the first letter of Long Name tokens). If no acronym expansion was detected within a group, we mapped the Short Name token to the shortest phrase in the Long Name group. This process resulted in a cross-walk from LOINC Short Name Tokens to LOINC Long Name tokens/phrases.

To handle abbreviations contained in the LOINC System field, we used string distance-matching with the Jaro-Winkler metric[49-51] to find the corresponding words with the smallest edit distance in the LOINC Long Name field. We mapped the System token to the resulting distance-matched Long Name token, except when the System could be mapped to an acronym expansion (in which case the acronym expansion was used).

LOINC Feature Engineering

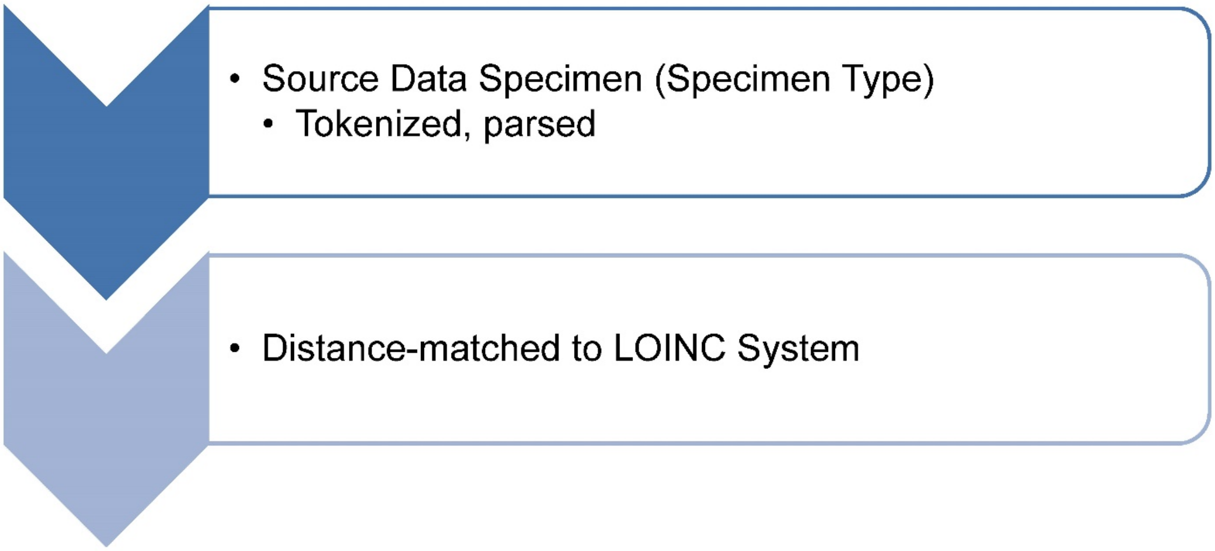
For the final model, we made no changes to the LOINC feature engineering methods presented in the initial model. We string distance-matched tokens from the processed source data test names and specimen types to tokens derived from the LOINC data preprocessing step. We used the Jaro-Winkler and Levenshtein distance metrics to identify the LOINC Long Name token with the smallest edit distance (**Figure 9C**).[52] For each test name and specimen type, we concatenated the resulting tokens from the Long Name to form the two ‘Test Name mapped to LOINC Long Name’ features (one for each distance-matching metric) and the two ‘Specimen Type mapped to LOINC Long Name’ features. We distance-matched the resulting mapped test names and specimen types, using both the Jaro-Winkler and Levenshtein metrics, to the LOINC Component and System fields, respectively. We included the predicted Component and System, along with their corresponding match distances from each of the two string distance-matching metrics, as model features. The process for mapping source data test names and specimen types are depicted in **Figure 10** and **Figure 11**. All features of the final model are shown in **Table 13**.

Figure 10. Test Name Text Processing



- Source Data Test Names
  - Tokenized, parsed
  - High-frequency terms eliminated using tunable threshold
- Distance-matched to LOINC Short Name Tokens
  - Mapped to corresponding LOINC Long Word/Phrase
  - Final mapped tokens concatenated to phrase
- Test Name phrase mapped to LOINC Component (by string distance-matching)

Figure 11. Specimen Type Text Processing



- Source Data Specimen (Specimen Type)
  - Tokenized, parsed
- Distance-matched to LOINC System

Feature Engineering Using UMLS CUIs

Using output from the automated text processing module, we leveraged the UMLS REST API to obtain UMLS CUIs for test names and specimen types, respectively (**Figure 9D**). In the initial step, we attempted to map the test name or specimen type to a UMLS CUI using the exact match search type. If no results were returned, we then attempted a ‘words’ search, in which a term is broken into its component parts and all concepts containing any words in the term are retrieved. Finally, if neither of the initial search types returned results, we iterated over the individual tokens in the test name and performed an exact match search for each token. For both the test name and the specimen type, if the resulting JSON object from a single UMLS search contained multiple CUIs, the first three CUIs were retained as model features (**Table 13**).

Table 13. Final Model Features

Text	Numeric
LOINC Long Name (JW) mapped from Test Name	Test result 5th percentile
LOINC Long Name (LV) mapped from Test Name	Test result 25th percentile
LOINC System (JW) mapped from Specimen Type	Test result median
LOINC System (LV) mapped from Specimen Type	Test result mean
Predicted LOINC Component (JW)	Test result 75th percentile
Component Match Distance (JW)	Test result 95th percentile
Predicted LOINC Component (LV)	Test result minimum
Component Match Distance (LV)	Test result maximum
Predicted LOINC System (JW)	Normalized test frequency*
System Match Distance (JW)	
Predicted LOINC System (LV)	
System Match Distance (LV)	
Units	
UMLS Test CUI #1	
UMLS Test CUI #2	
UMLS Test CUI #3	
UMLS Specimen CUI #1	
UMLS Specimen CUI #2	
UMLS Specimen CUI #3	

JW, Jaro-Winkler; LV, Levenshtein.

* Normalized test frequency calculation $= \frac{\text{Test frequency}}{\text{Total number of laboratory results per site}}$

Data Filtering and Partitioning

We held out instances with missing specimen type and/or LOINC code in the unlabeled dataset for a separate analysis. We combined data instances containing LOINC codes used at only one site or <10 times by test frequency with the unlabeled dataset for reclassification. In the remaining labeled dataset, we partitioned data for 5-fold cross-validation using splits by sites.

Automating LOINC Equivalence and Forming LOINC Groups

Based on the challenges identified in our initial model iteration, we automated the creation of equivalent LOINC code groups (see Chapter 2 for full details). Using the LOINC group keys defined above, all LOINC codes present in the original source data were 'rolled up' into corresponding LOINC keys where possible and appended to the analytic data file. If the original LOINC codes were not part of an interoperable LOINC group, the original LOINC code was retained as the LOINC key.

Machine Learning Models

We implemented logistic regression (L1 penalized,[58] L2 penalized,[59] and L1/L2 penalized[60]), a random forest[61] multiclass classifier, and a one-versus-rest ensemble of binary random forest classifiers. Model building and analyses were conducted using scikit-learn in Python.[53] We tuned all models with 5-fold cross-validation using the weighted F1 score as the loss function. For the logistic regression models, we tuned parameters using grid search. Because the random forest models have multiple hyperparameters to tune which makes an exhaustive grid search infeasible, we tuned the following random forest hyperparameters using the hyperopt package:[62] criterion (function to measure the quality of a split), number of estimators (numbers of trees in the forest), maximum features (number of features to consider when looking for the best split), maximum tree depth, and minimum samples per split.

Estimating Model Performance

Using 5-fold cross-validation by site in the labeled dataset, we estimated performance for each model with the following measures: accuracy, weighted F1 score, and micro-averaged F1 score. Accuracy represents the number of correct labels divided by the number of instances. The F1 score is the harmonic mean of precision (positive predictive value) and recall (sensitivity).[63] In the weighted F1 score, the metrics for each label are calculated, averaged, and weighted by the support (number of true instances for each label).[53] In the micro-averaged F1 score, metrics are calculated globally by counting the total true positive, false negatives, and false positives.[53] We included accuracy for intuitive interpretation. Since accuracy can be optimistic with class imbalance (simply predicting the labels of the most common classes), we examined the weighted F1 score and the micro-averaged F1 score. We also calculated expected accuracy with random guessing in proportion to label prevalence.

Within each of the three measures (accuracy, weighted F1, and micro-averged F1), we evaluated performance differences among the five models using a one-way anlysis of variance (ANOVA),[64] followed by independent two-sample t-tests[65] between each pair of models when findings from the ANOVA test were significant ($p < 0.05$). We also calculated 95% confidence intervals for the performance measures of each of the five models using their mean and standard deviation from the 5-fold cross-validation.

Model Fitting and Label Assignment

The model has 2 potential use cases: 1) predicting labels when new sites are added to an existing model, and 2) reclassifying incorrect labels in retrospective multi-site data. In the first use case, cross-validated performance is of interest to estimate how the model would perform if data from new sites were added. In the second case, overfitting is not an issue, obviating the need to estimate performance with cross-validation. To examine these two use cases, we fit the best-performing model to the training data during cross-validation (CV Model) and to the full labeled dataset (Full Model). For the CV Model and the Full Model, we obtained the predicted LOINC keys as described above. In cases where the predicted LOINC code was not identical to the original LOINC code, but the predicted LOINC code was the key for the group containing the original LOINC code, we retained the original LOINC code. When the predicted code was not interoperable with the original LOINC code, we retained the predicted LOINC code. Subsequently, to evaluate model utility for mapping data with missing labels or instances originally labeled with infrequently used LOINC codes, the CV Model and the Full Model were used to predict LOINC codes on the holdout unlabeled dataset.

Manual Validation

We performed manual validation on random samples from both the labeled and unlabeled datasets. Within each dataset, we selected two instances from each of the 130 sites. Using the cumulative sum of test frequency within a site, we selected one instance with test frequency ≥50%, and one instance with test frequency <50% (**Table 14**). Descriptions of the adjudication label categories are shown in **Tables 15-17**. We examined the accuracy of the labels predicted in the CV Model and the Full Model. Additionally, to explicitly evaluate model utility for reclassifying incorrect LOINC codes in the dataset, we obtained a sample of 260 instances in the labeled dataset where the predicted LOINC code (Full Model) differed from the original source data LOINC code. Two reviewers (one physician [SKP] and one nurse practitioner [ADJ]) manually reviewed a total of 780 records. We report the inter-annotator agreement using Cohen’s kappa. In the case of adjudication disagreement, we used consensus agreement to determine the final adjudication.

Table 14. Manual Validation Data Sampling Strategy

	Unlabeled Data	Randomly-Sampled Labeled Data	Discordant Labeled Data
Top 50% by cumulative sum of testing frequency	130 records	130 records	130 records
Bottom 50% by cumulative sum of testing frequency	130 records	130 records	130 records

In the unlabeled dataset, there were 5 possible mutually exclusive labels. These labels and their definitions are shown in **Table 15**.

Table 15. Label Categories for Manual Validation in Unlabeled Dataset

Label	**Definition**
Predicted Correct	Model-predicted label is correct
Predicted Incorrect	Model-predicted label is incorrect
Insufficient Or Conflicting Information	Either not enough source data to infer code (i.e. units missing and would be necessary to assign code), or source data conflicts (i.e. test name includes the word 'blood' and specimen type is 'urine')
No LOINC Coverage, Code Synonymous	LOINC code does not exist for the combination of test and specimen type in the source data, but the predicted LOINC code is the most reasonable alternative (i.e. protein, blood does not exist in LOINC; protein, serum/plasma is the reasonable alternative)
No LOINC Coverage, Code Incorrect	LOINC code does not exist for the combination of test and specimen type in the source data, and the predicted LOINC code is not a reasonable alternative (i.e. protein, blood does not exist in LOINC; protein, urine is not a reasonable alternative)

In the randomly-sampled labeled dataset, there were 9 possible mutually exclusive labels after fitting the model to the full dataset. These labels and their definitions are shown in **Table 16.**

Table 16. Label Categories for Manual Validation in the Randomly-Sampled Labeled Dataset

Label	**Definition**
Concordant Correct	Model-predicted label = original label and is correct
Concordant Incorrect	Model-predicted label = original label and is incorrect
Discordant Predicted Correct	Model-predicted label ≠ original label and model-predicted label is correct
Discordant Original Correct	Model-predicted label ≠ original label and original label is correct
Discordant Neither Correct	Model-predicted label ≠ original label and neither label is correct
Discordant Both Correct	Model-predicted label ≠ original label and both labels are correct (equivalent)
Insufficient Or Conflicting Information	Either not enough source data to infer code (i.e. specimen type cannot be extrapolated to something meaningful), or source data conflicts (i.e. test name includes the word 'blood' and specimen type is urine)
No LOINC Coverage Code Correct Synonymous	LOINC code does not exist for the combination of test and specimen type in the source data, but the predicted LOINC code is the most reasonable alternative (i.e. protein, blood does not exist in LOINC; protein, serum/plasma is the reasonable alternative)
No LOINC Coverage Code Incorrect	LOINC code does not exist for the combination of test and specimen type in the source data, and the predicted LOINC code is not a reasonable alternative (i.e. protein, blood does not exist in LOINC; protein, urine is not a reasonable alternative)

In the targeted evaluation of the labeled dataset where the model-predicted LOINC key differed from the original source data LOINC key, there were 7 potential labels, listed with their definitions in **Table 17**.

Table 17. Label Categories for Manual Validation in Labeled Dataset where Original and Predicted LOINC Codes Disagree (Discordant)

Label	**Definition**
Predicted Correct	Model-predicted label correct (original label incorrect)
Both Correct (Synonymous)	Model-predicted label and original label correct but not captured by LOINC equivalence algorithm logic
Original Correct	Original label correct (model-predicted label incorrect)
Neither Correct	Model-predicted label and original label both incorrect
Insufficient Or Conflicting Information	Either not enough source data to infer code (i.e. units missing and would be necessary to assign code), or source data conflicts (i.e. test name includes the word 'blood' and specimen type is 'urine')
No LOINC Coverage, Code Synonymous	LOINC code does not exist for the combination of test and specimen type in the source data, but the predicted LOINC code is the most reasonable alternative (i.e. protein, blood does not exist in LOINC; protein, serum/plasma is the reasonable alternative)
No LOINC Coverage, Code Incorrect	LOINC code does not exist for the combination of test and specimen type in the source data, and the predicted LOINC code is not a reasonable alternative (i.e. protein, blood does not exist in LOINC; protein, urine is not a reasonable alternative)

Examining Model Performance by Dataset Characteristics

From the full labeled dataset, we randomly sampled between 5 and 125 sites (in increments of 5 sites) and fit a random forest multiclass classifier with 5-fold cross-validation split by sites to assess model performance. Within each sampled data subset, we calculated the number of distinct LOINC keys and the number of data instances, and examined their relationship with model performance.

All source code was developed in Python 3.6.0, and is available at https://github.com/skparr/ml_loinc_mapping. For string distance matching, we used the R stringdist package[66,67] within Python via the rpy2 package.[68] **Table 18** contains tool options that can be parameterized to provide flexibility for the user. Once the user specifies the variables in the configuration file (detailed in the README.md file), the program can be run via command line execution of a single Python script.

Table 18. Parameterizable Tool Options

Frequency threshold for uninformative token elimination

Number of CUIs returned from the test name and specimen search via the UMLS API

Data filtering criterion (i.e. LOINC codes present at only one site, or LOINC codes used less than a certain threshold value by test volume)

Granularity of hyperparameter grid

Number of trials to be attempted using the hyperopt hyperparameter tuning package

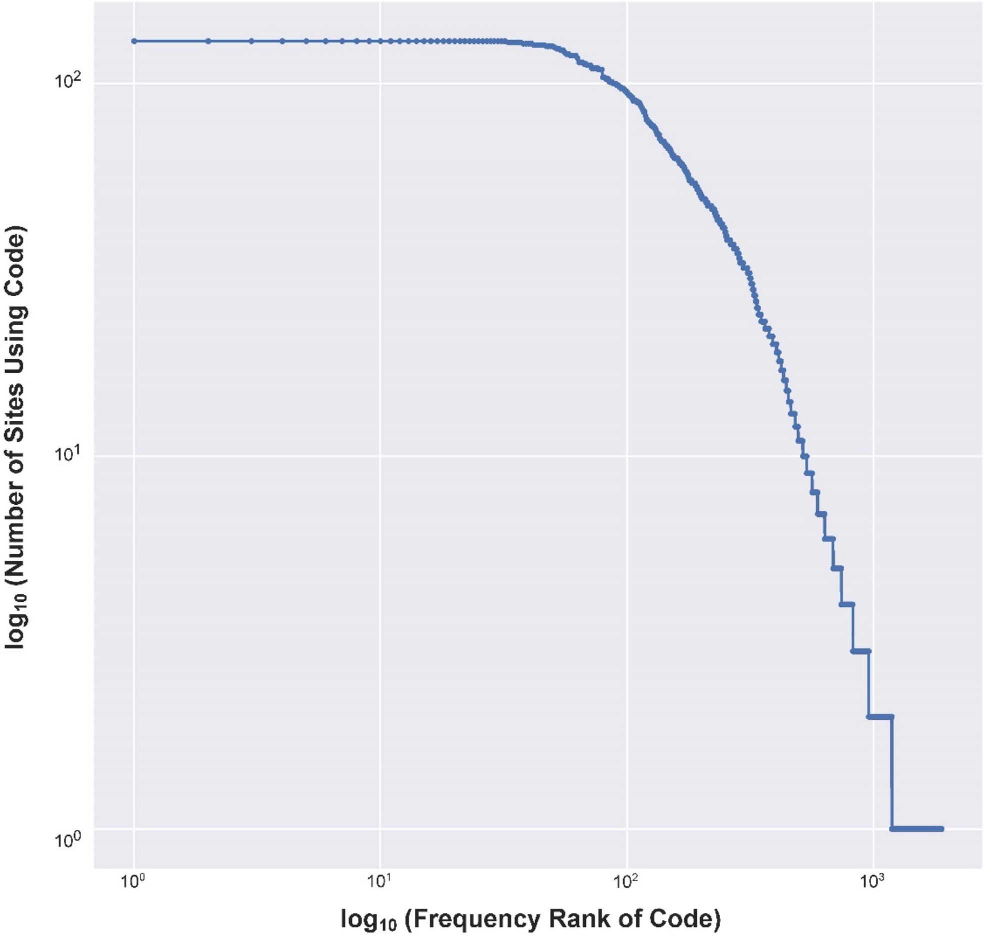
Number of cross-validation folds used in both hyperparameter tuning and estimation of model performance

Results

The raw laboratory data consisted of over 6.5 billion test results, ranging from 2.5 to 184 million results per site (median 41.2 million). After aggregating by laboratory test identifier, specimen type identifier, units, and LOINC code, the analytic dataset consisted of 140,565 instances and 2,215 distinct LOINC codes. LOINC codes were missing in 41,301 source data instances (29%), corresponding to 450 million test results.

Of the 1,895 distinct LOINC keys remaining after grouping, less than 100 keys were used consistently across sites (top left, **Figure** 12), and many were used at less than 10 sites (bottom right, **Figure** 12). The 707 keys used at only a single site and the 24 keys used fewer than 10 times by total test frequency were combined with the unlabeled data for reclassification.

Figure 12. LOINC Code Frequency Rank versus Number of Sites using Code



The filtered, labeled dataset consisted of 94,845 data instances, aggregated from approximately 6.1 billion individual test results, with 1,164 distinct LOINC keys. The dataset comprised of unlabeled and/or infrequent tests consisted of 42,720 instances, aggregated from approximately 462 million individual test results.

Cross-Validated Model Performance

The random forest models (one-versus-rest and multiclass) significantly outperformed the three logistic regression models in all performance measures (**Table 19**). All models performed considerably better than random guessing in proportion to the prevalence of the

1,164 possible class labels, which would yields an accuracy of 0.5%. The random forest classifiers in the final model performed significantly better than the model prototype presented in Chapter 2 (RF multiclass: 63.8% vs 56.8% and RF one-versus-rest 64.9% vs 57.7%).

Table 19. Final Model Cross-validated Performance

	Accuracy (95% CI)	Weighted F1 Score (95% CI)	Micro-Averaged F1 (95% CI)
L1	0.568 (0.559-0.578)	0.551 (0.537-0.565)	0.568 (0.559-0.578)
L2	0.606 (0.591-0.621)	0.556 (0.536-0.577)	0.606 (0.591-0.621)
L1-L2	0.607 (0.593-0.621)	0.562 (0.543-0.582)	0.607 (0.593-0.621)
RF (multiclass)	**0.638** (0.622-0.653)*	**0.612** (0.594-0.630)*	**0.638** (0.623-0.654)*
RF (one-versus-rest)	**0.649** (0.632-0.666)*	**0.621** (0.601-0.640)*	**0.649** (0.632-0.666)*

Abbreviations: CI, Confidence Interval. L1, L1 penalized logistic regression; L2, L2 penalized logistic regression; L1-L2, L1-L2 penalized logistic regression; RF, random forest.
* P-values <0.05 within each of the three performance measures for comparisons between RF (multiclass) and, L1, L2, and L1-L2 LR models and for comparisons between RF (one-versus-rest) and, L1, L2, and L1-L2 LR models.

Manual Validation

Full Model

Unlabeled Data

Using the Full Model applied to the unlabeled data, Cohen's kappa for inter-rater agreement was 0.76 (**Table 20**). The model-predicted label was correct in 84.7% of records by test frequency. Model performance by test frequency was comparable in the infrequent (Bottom 50%) and frequent (Top 50%) tests, but by instance the model performed better on the frequent tests (**Table 21**).

Table 20. Adjudicator Agreement on Manual Validation with Model Fit to Full Labeled Dataset

	% Agreement	**Cohen's kappa**
Unlabeled Data	88.1%	0.76
Randomly-Sampled Labeled Data	92.7%	0.82
Discordant Labeled Data	84.6%	0.70

Table 21. Manual Validation in Unlabeled Data (Model Fit to Full Labeled Dataset)

	Unlabeled Data					
	Bottom 50%		**Top 50%**		**Total**	
	Instances (N=130)	**Tests (N=944,156)**	**Instances (N=130)**	**Tests (N=30,776,801)**	**Instances (N=260)**	**Tests (N=31,720,957)**
Total Correct	87 (66.9%)	798,268 (84.5%)	108 (83.1%)	26,054,265 (84.7%)	195 (75.0%)	26,852,533 (84.7%)
Predicted Correct	70 (53.9%)	599,043 (63.4%)	106 (81.5%)	25,910,603 (84.2%)	176 (67.7%)	26,509,646 (83.6%)
No LOINC Coverage, Code Synonymous	17 (13.1%)	199,225 (21.1%)	2 (1.5%)	143,662 (0.5%)	19 (7.3%)	342,887 (1.1%)
Total Incorrect	26 (20%)	114,632 (12.1%)	19 (14.6%)	4,285,372 (13.9%)	45 (17.3%)	4,400,004 (13.9%)
Predicted Incorrect	22 (16.9%)	114,622 (12.1%)	19 (14.6%)	4,285,372 (13.9%)	41 (15.8%)	4,399,994 (13.9%)
No LOINC Coverage, Code Incorrect	4 (3.1%)	10 (<0.1%)	0 (0%)	0 (0%)	4 (1.5%)	10 (<0.1%)
Insufficient or Conflicting Information	17 (13.1%)	31,256 (3.3%)	3 (2.3%)	437,164 (1.4%)	20 (7.7%)	468,420 (1.5%)

Definitions: Predicted Correct: Model-predicted label is correct; No LOINC Coverage, Code Synonymous: LOINC code does not exist for the combination of test and specimen type in the source data, but the predicted LOINC code is the most reasonable alternative; Predicted Incorrect: Model-predicted label is incorrect; No LOINC Coverage, Code Incorrect: LOINC code does not exist for the combination of test and specimen type in the source data, and the predicted LOINC code is not a reasonable alternative; Insufficient or Conflicting Information: Either not enough source data to infer code (i.e. units missing and would be necessary to assign code), or source data conflicts (i.e. test name includes the word ‘blood’ and specimen type is ‘urine’).

Randomly-Sampled Labeled Data

In the labeled dataset, inter-rater agreement assessed by Cohen’s kappa was 0.82 (**Table 20**). The model-predicted label was correct in 95.9% of records by test frequency, with higher accuracy in the frequent tests than in infrequent tests (**Table 22**).

Table 22. Manual Validation in Randomly-sampled Labeled Data (Model Fit to Full Labeled Dataset)

	Randomly-Sampled Labeled Data					
	Bottom 50%		**Top 50%**		**Total**	
	Instances (N=130)	**Tests (N=4,678,607)**	**Instances (N=130)**	**Tests (N=136,643,970)**	**Instances (N=260)**	**Tests (N=141,322,577)**
Total Correct	81 (62.3%)	3,801,382 (81.3%)	126 (96.9%)	131,790,613 (96.4%)	207 (79.6%)	135,591,995 (95.9%)
Concordant Correct	71 (54.6%)	3,763,546 (80.4%)	124 (95.4%)	129,207,143 (94.6%)	195 (75%)	132,970,689 (94.1%)
Discordant Predicted Correct	7 (5.4%)	37,612 (0.8%)	1 (0.8%)	1,565,720 (1.1%)	8 (3.1%)	1,603,332 (1.1%)
No LOINC Coverage, Code Synonymous	3 (2.3%)	224 (<0.1%)	1 (0.8%)	1,017,750 (0.7%)	4 (1.5%)	1,017,974 (0.7%)
Total Incorrect	31 (23.8%)	876,859 (18.7%)	4 (3.1%)	4,853,357 (3.6%)	35 (13.5%)	5,730,216 (4.1%)
Concordant Incorrect	25 (19.2%)	876,829 (18.7%)	3 (2.3%)	2,782,119 (2.0%)	28 (10.8%)	3,658,948 (2.6%)
Discordant Original Correct	1 (0.8%)	1 (<0.1%)	1 (0.8%)	2,071,238 (1.5%)	2 (0.8%)	2,071,239 (1.5%)
Discordant Neither Correct	1 (0.8%)	15 (<0.1%)	0 (0%)	0 (0%)	1 (0.4%)	15 (<0.1%)
No LOINC Coverage, Code Incorrect	4 (3.1%)	14 (<0.1%)	0 (0%)	0 (0%)	4 (1.5%)	14 (<0.1%)
Insufficient or Conflicting Information	18 (13.8%)	366 (<0.1%)	0 (0%)	0 (0%)	18 (6.9%)	366 (<0.1%)

Definitions: Concordant Correct: Model-predicted label = original label and is correct; Discordant Predicted Correct: Model-predicted label ≠ original label and model-predicted label is correct; No LOINC Coverage, Code Synonymous: LOINC code does not exist for the combination of test and specimen type in the source data, but the predicted LOINC code is the most reasonable alternative; Concordant Incorrect: Model-predicted label = original label and is incorrect; Discordant Original Correct: Model-predicted label ≠ original label and original label is correct; Discordant Neither Correct: Model-predicted label ≠ original label and neither label is correct; Insufficient or Conflicting Information: Either not enough source data to infer code (i.e. units missing and would be necessary to assign code), or source data conflicts (i.e. test name includes the word ‘blood’ and specimen type is ‘urine’).

Targeted Evaluation of Discordant Labels

In manual validation of cases where the LOINC code present in the source data (original label) differed from the model-predicted LOINC code, Cohen's kappa for inter-rate agreement was 0.70 (**Table 20**). The model-predicted LOINC code was correct in 83.2% by test frequency, and the model-predicted LOINC code was better than the original label 71.5% of the time by test frequency (**Table 23**).

Table 23. Manual Validation in Labeled Data where Original and Predicted LOINC Codes Discordant (Model Fit to Full Labeled Dataset)

	Discordant Labeled Data					
	Bottom 50%		**Top 50%**		**Total**	
	Instances (N=130)	**Tests (N=593,709)**	**Instances (N=130)**	**Tests (N=48,825,571)**	**Instances (N=260)**	**Tests (N=49,419,280)**
Total Correct	87 (66.9%)	532504 (89.7%)	117 (90%)	40570921 (83.1%)	204 (78.5%)	41103425 (83.2%)
Predicted Correct	77 (59.2%)	501167 (84.4%)	108 (83.1%)	34857234 (71.4%)	185 (71.2%)	35358401 (71.5%)
Both Correct (Synonymous)	1 (0.8%)	31153 (5.2%)	6 (4.6%)	5252216 (10.8%)	7 (2.7%)	5283369 (10.7%)
No LOINC Coverage, Code Synonymous	9 (6.9%)	184 (0%)	3 (2.3%)	461471 (0.9%)	12 (4.6%)	461655 (0.9%)
Total Incorrect	33 (25.4%)	61157 (10.3%)	11 (8.5%)	7478321 (15.3%)	44 (16.9%)	7539478 (15.3%)
Original Correct	15 (11.5%)	20121 (3.4%)	8 (6.2%)	7023988 (14.4%)	23 (8.8%)	7044109 (14.3%)
Neither Correct	18 (13.8%)	41036 (6.9%)	3 (2.3%)	454333 (0.9%)	21 (8.1%)	495369 (1%)
Insufficient or Conflicting Information	10 (7.7%)	48 (0%)	2 (1.5%)	776329 (1.6%)	12 (4.6%)	776377 (1.6%)

Definitions: Predicted Correct: Model-predicted label correct (original label incorrect); Both Correct (Synonymous): Model-predicted label and original label correct; No LOINC Coverage, Code Synonymous: LOINC code does not exist for the combination of test and specimen type in the source data, but the predicted LOINC code is the most reasonable alternative; Original Correct: Original label correct (model-predicted label incorrect); Neither Correct: Model-predicted label and original label both incorrect; Insufficient Or Conflicting Information: Either not enough source data to infer code (i.e. units missing and would be necessary to assign code), or source data conflicts (i.e. test name includes the word ‘blood’ and specimen type is ‘urine’).

CV Model

Unlabeled Data

Using the CV Model applied to the unlabeled dataset, Cohen's kappa for inter-rater agreement was 0.73 (**Table 24**). The model-predicted label was correct in 82.3% of records by test frequency, which is similar to the results from the Full Model. Compared to the Full Model, the CV Model performed modestly better in the infrequent tests and slightly worse in the frequent tests (**Table 21 and Table 25**).

Table 24. Adjudicator Agreement on Manual Validation with Model Fit During 5-fold Cross-Validation

	% Agreement	**Cohen's kappa**
Unlabeled Data	86.9%	0.73
Randomly-Sampled Labeled Data	93.4%	0.86

Table 25. Manual Validation in Unlabeled Data (Model Fit During 5-fold Cross-validation)

	Unlabeled Data					
	Bottom 50%		Top 50%		Total	
	Instances (N=130)	Tests (N=944,156)	Instances (N=130)	Tests (N=30,776,801)	Instances (N=260)	Tests (N=31,720,957)
Total Correct	89 (68.5%)	900,072 (95.3%)	106 (81.5%)	25,212,942 (81.9%)	195 (75%)	26,113,014 (82.3%)
Predicted Correct	72 (55.4%)	700,972 (74.2%)	104 (80.0%)	25,069,280 (81.5%)	176 (67.7%)	25,770,252 (81.2%)
No LOINC Coverage, Code Synonymous	17 (13.1%)	199,100 (21.1%)	2 (1.5%)	143,662 (0.5%)	19 (7.3%)	342,762 (1.1%)
Total Incorrect	24 (18.5%)	12,828 (1.4%)	21 (16.2%)	5,126,695 (16.7%)	45 (17.3%)	5,139,523 (16.2%)
Predicted Incorrect	21 (16.2%)	12,822 (1.4%)	21 (16.2%)	5,126,695 (16.7%)	41 (15.8%)	5,139,517 (16.2%)
No LOINC Coverage, Code Incorrect	3 (2.3%)	6 (<0.1%)	0 (0%)	0 (0%)	4 (1.5%)	6 (<0.1%)
Insufficient or Conflicting Information	17 (13.1%)	31,256 (3.3%)	3 (2.3%)	437,164 (1.4%)	20 (7.7%)	468,420 (1.5%)

Definitions: Predicted Correct: Model-predicted label is correct; No LOINC Coverage, Code Synonymous: LOINC code does not exist for the combination of test and specimen type in the source data, but the predicted LOINC code is the most reasonable alternative; Predicted Incorrect: Model-predicted label is incorrect; No LOINC Coverage, Code Incorrect: LOINC code does not exist for the combination of test and specimen type in the source data, and the predicted LOINC code is not a reasonable alternative; Insufficient or Conflicting Information: Either not enough source data to infer code (i.e. units missing and would be necessary to assign code), or source data conflicts (i.e. test name includes the word 'blood' and specimen type is 'urine').

Randomly-Sampled Labeled Data

Cohen’s kappa for inter-rater agreement was 0.86 in the labeled dataset (**Table 24**). The model-predicted label was correct in 94.8% of records by test frequency, which is similar to the results from the Full Model. Compared to the Full Model, incorrect predictions in the CV Model were driven by more instances in which the original and predicted labels disagreed and were both incorrect (Discordant Neither Correct), but fewer instances in which the original and predicted labels agreed and were incorrect (Concordant Incorrect) (**Table 22 and Table 26**).

Table 26. Manual Validation in Randomly-sampled Labeled Data (Model Fit During 5-fold Cross-validation)

	Randomly-Sampled Labeled Data					
	Bottom 50%		Top 50%		Total	
	Instances (N=130)	Tests (N=4,678,607)	Instances (N=130)	Tests (N=136,643,970)	Instances (N=260)	Tests (N=141,322,577)
Total Correct	81 (62.3%)	3,508,110 (75%)	125 (96.2%)	130,461,944 (95.5%)	206 (79.2%)	133,970,054 (94.8%)
Concordant Correct	61 (46.9%)	3,452,906 (73.8%)	123 (94.6%)	127,878,474 (93.6%)	184 (70.8%)	131,331,380 (92.9%)
Discordant Predicted Correct	13 (10%)	54,966 (1.2%)	1 (0.8%)	1,565,720 (1.1%)	14 (5.4%)	1,620,686 (1.1%)
No LOINC Coverage, Code Synonymous	7 (5.4%)	238 (<0.1%)	1 (0.8%)	1,017,750 (0.7%)	8 (3.1%)	1,017,988 (0.7%)
Total Incorrect	31 (23.8%)	1,170,131 (25.0%)	5 (3.8%)	6,182,026 (4.5%)	36 (13.8%)	7,352,157 (5.2%)
Concordant Incorrect	10 (7.7%)	239,959 (5.1%)	3 (2.3%)	2,782,119 (2.0%)	13 (5.0%)	3,022,078 (2.1%)
Discordant Original Correct	1 (0.8%)	1 (<0.1%)	1 (0.8%)	2,071,238 (1.5%)	2 (0.8%)	2,071,239 (1.5%)
Discordant Neither Correct	20 (15.4%)	930,171 (19.9%)	1 (0.8%)	1,328,669 (1%)	21 (8.1%)	2,258,840 (1.6%)
No LOINC Coverage, Code Incorrect	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
Insufficient or Conflicting Information	18 (13.8%)	366 (<0.1%)	0 (0%)	0 (0%)	18 (6.9%)	366 (<0.1%)

Definitions: Concordant Correct: Model-predicted label = original label and is correct; Discordant Predicted Correct: Model-predicted label ≠ original label and model-predicted label is correct; No LOINC Coverage, Code Synonymous: LOINC code does not exist for the combination of test and specimen type in the source data, but the predicted LOINC code is the most reasonable alternative; Concordant Incorrect: Model-predicted label = original label and is incorrect; Discordant Original Correct: Model-predicted label ≠ original label and original label is correct; Discordant Neither Correct: Model-predicted label ≠ original label and neither label is correct; Insufficient or Conflicting Information: Either not enough source data to infer code (i.e. units missing and would be necessary to assign code), or source data conflicts (i.e. test name includes the word ‘blood’ and specimen type is ‘urine’).

Estimated Noisy Label Prevalence

In manual validation of randomly-sampled labeled data, noisy labels (incorrect labels in the original source data) are comprised of the Discordant Predicted Correct, Discordant Neither Correct, and Concordant Incorrect categories in **Table 22**. Considering the 234 instances in which LOINC coverage existed for the source data, and in which there was sufficient information to determine the LOINC code, the noisy label prevalence is 15.8% (**Table 27**). In this data sample, if all original labels were replaced with the model-predicted labels, the error rate would be 13.2%.

Table 27. Estimating Noisy Label Prevalence from Randomly-sampled Labeled Data (Full Model)

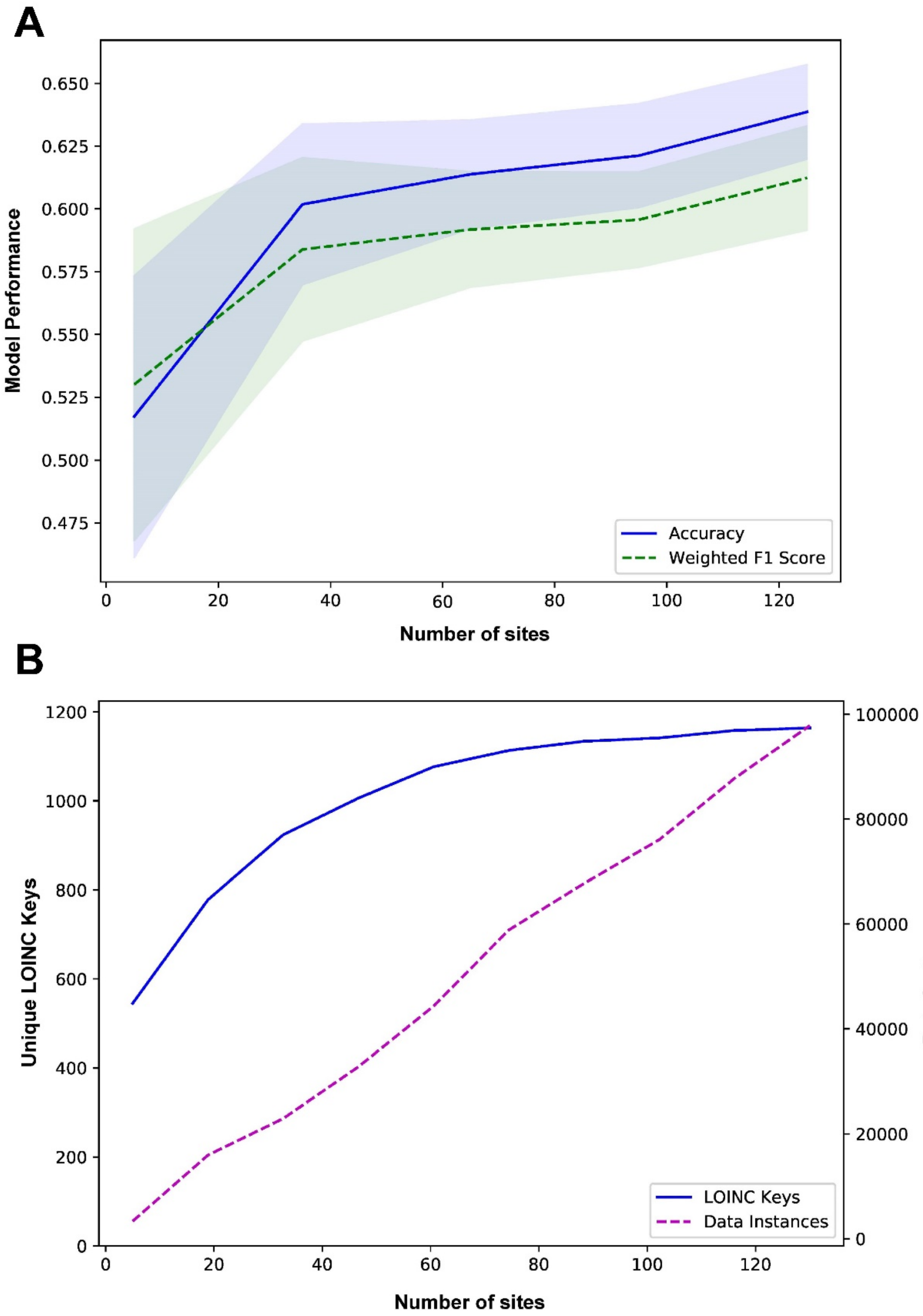
	Instances (N=234)
Noisy Labels	37 (15.8%)
Concordant Incorrect	28 (12.0%)
Discordant Predicted Correct	8 (3.4%)
Discordant Neither Correct	1 (0.4%)
Correct Labels	197 (84.2%)
Concordant Correct	195 (83.3%)
Discordant Original Correct	2 (0.9%)

Full Model refers to the One-Versus-Rest classifier fit to the full labeled dataset.
Label Definitions: Concordant Incorrect: Model-predicted label = original label and is incorrect; Discordant Predicted Correct: Model-predicted label ≠ original label and model-predicted label is correct; Discordant Neither Correct: Model-predicted label ≠ original label and neither label is correct; Concordant Correct: Model-predicted label = original label and is correct; Discordant Original Correct: Model-predicted label ≠ original label and original label is correct.

Examining Model Performance by Dataset Characteristics

When the number of sites in the model ranged from 5 to 35, performance improved dramatically with the addition of data in 5-site increments (**Figure 13.A**). Increasing the number of sites beyond 35 (up to 125) provided modest, albeit continued, performance improvement. The number of unique LOINC keys in the data also increased most appreciably in the range of 5 to 35 sites, plateauing when approximately 80 sites were included in the model (**Figure 13.B**). As sites were added to the dataset, the number of data instances increased linearly across the entire range.

Figure 13. Examining Model Performance by Dataset Characteristics.



A) Model accuracy (solid line) and weighted F1 score (dashed line) with 95% confidence bands for random forest multiclass model fit in 5-fold cross-validation to randomly sampled data subsets with varying number of sites. **B)** Number of unique LOINC keys (solid line) and number of data instances (dashed line) in randomly sampled data subsets with varying number of sites.

CHAPTER 4

DISCUSSION

In this study we automated feature generation and mapping of laboratory data to LOINC codes using a series of automated data transformation modules and a machine learning algorithm that leverages noisy labels within a large, heterogeneous national electronic health record system database. Using this method, we were able to assign LOINC codes to unlabeled data with reasonable performance. We demonstrated comparable label accuracy when the model was fit to the entire dataset or when labels were assigned during cross-validation, suggesting that this model could be used on existing retrospective datasets or applied to new sites. Additionally, our model demonstrated utility in LOINC code reclassification, which could serve to augment data quality.

Our results are similar in accuracy to the best reported methods that have previously attempted to automate laboratory mapping.[20-22] Notably, our estimates of model performance may actually be conservative for two main reasons. First, we did not exclude tests that occurred rarely (i.e. <10 results during the 16-year data collection timeframe). We attempted to map these results both for generalizability and to assess model performance with rare occurrences. Second, during manual validation we did not consider clinical equivalence in determining label accuracy. For example, using the LOINC Groups classification,[56] a test for Glucose [Mass/volume] in Capillary blood (LOINC code 32016-8) and a test for Glucose [Mass/volume] in Blood (LOINC code 2339-0) can be grouped by the parent code LG11181-1. However, we considered a label of 2339-0 for the test 'Glucose, Capillary Blood' incorrect, because a model would ideally assign the more specific code 32016-8 given the information in the source data. We opted to stringently assess model label accuracy, because an ideal model would assign the most granular label that represents the data and allow the end-user to aggregate codes if desired. We chose not use the publicly available LOINC Multi-Axial Hierarchy table, which in some cases groups LOINC codes with differing Property and Scale. For example, tests with quantitative results may be grouped with tests reported in ordinal scale. Since we aimed to map laboratory tests in a way that would not require the end-user to filter, sort, or transform tests within a LOINC group, we used the LOINC equivalence algorithm detailed in the Methods section.

In our manual validation, the model performed better within the labeled data than the unlabeled data. This is not surprising, given that unlabeled data is not necessarily unlabeled at random, and contains an over-representation of unusual combinations of test name, specimen type, and/or units. Nonetheless, within unlabeled data, the model-predicted LOINC code was correct in 85% of the sampled data by test frequency. Depending on the use case, the gain in useable data provided by assigning LOINC codes to unlabeled data (albeit, with some incorrect labels) might offset the misclassification rate. We observed that in both the labeled and unlabeled data, the model performed better within common tests (Top 50%) than within uncommon tests (Bottom 50%). This is not unexpected, given that uncommon tests also contain unusual combinations of test name, specimen type, and/or units. Unlike previous studies, we attempted to map all data instances for generalizability, but our results suggest that model performance (and confidence in label predictions) could be improved by restricting to common tests. Where the model-predicted LOINC code disagreed with the original source data LOINC code, the model-predicted code was correct *and* more appropriate than the original code in 71% of instances. Implementation of our current model without any modification would still generally improve the original source data quality by correctly reclassifying LOINC codes in 72% of the data by test frequency, while incorrectly reclassifying 14% of labels whose original assignment was correct.

In this study, the random forest models outperformed penalized logistic regression models, which is not surprising given that random forests are inherently multi-class capable and robust to label noise.[61] Additionally, random forest models are attractive because they automatically handle non-linear relationships and high-order variable interactions, and do not require binary expansion of categorical variables or standardization of continuous variables.

Strengths

Strengths and novelties of this study include: (a) use of a large (6.6 billion laboratory results) heterogeneous data source (130 sites) for model development, (b) implementation of an automated pipeline, (c) generalizable application, and (d) leveraging of noisy labels.

Automated Pipeline

Prior to our study, there have been no truly automated methods to map laboratory tests to LOINC codes. Previous 'automated' methods required manual work by domain experts, either to extensively map local terms to LOINC codes (corpus-based methods), or to choose the correct mapping from a list of candidates generated by the mapping tool (lexical method). The method we present fully automates the following steps: source data text processing and normalization, acronym and abbreviation expansion, synonym detection, feature engineering, and mapping/LOINC code assignment. The tool we present only requires the user to supply their aggregate laboratory source data and to enter the following mandatory fields into the configuration file: input and output file directories, the R library location, the user UMLS api key, and the names of the variables in the user's laboratory data file. Once the user specifies the configuration variables, the program can be run via command line execution of a single Python script. In the final output for both labeled and unlabeled data, the model-predicted LOINC code is appended to the source data file and written to the output directory.

Generalizability

The methods we present are generalizable to laboratory data at any healthcare institution and are not dependent on any proprietary VA laboratory information. To implement our model, the laboratory data need only contain a laboratory test name, specimen type, and a numeric test result, standard elements for laboratory data in any electronic health record system. We designed the model to handle data that contains only partial LOINC mappings and missing fields.

Noisy Labels

By including noisy labels, model performance will be inherently dependent upon the quality of the underlying labels. In this data source with an estimated noisy label rate of 16%, model performance was reasonable and prior research suggests that higher rates of noisy labels may be tolerated by machine learning methods. By incorporating noisy labels, we obviate the need for the manual corpus adjudication used in prior studies.[20,22] This is important because for corpus-based methods to be generalizable, the corpus must be large and heterogeneous, which requires significant and potentially non-scalable upfront manual effort to extensively map local laboratory tests. We developed our model using a large, heterogeneous data source, but because we leveraged noisy labels, no manual effort was required.

Limitations

Our study is not without limitations. First, because this model was developed using a large, national data source, our approach may not be generalizable to organizations with fewer sites. However, in our sensitivity analysis examining model performance by varying dataset characteristics, performance was reasonable with approximately 35 sites. Furthermore, performance appears to be more closely correlated with the number of distinct LOINC codes in the dataset rather than the number of data instances, suggesting that the model might perform well even in smaller organizations with heterogeneous data. Second, we restricted to the 150 most common laboratory tests with numeric results at each site, which could limit generalizability. However, the model may continue to perform well with addition of more tests due to our study design. First, because the top 150 tests were not identical across all of the 130 sites, the data used to train and evaluate the model was heterogeneous. Second, the 150 most common tests per site were selected based upon the local laboratory test name, but those test names could be associated with different specimen types and/or units, resulting in 219 to 2153 distinct combinations of test name/specimen type/units per site. Third, for our manual validation, we sampled from both commonly- (Top 50%) and uncommonly- (Bottom 50%) used tests. We used this sampling strategy to explicitly examine how the model performs with rarer data occurrences. Because we used heterogeneous data with rare occurrences for model development, the model may perform well with addition of more tests. Another potential limitation is that our model uses LOINC keys, which effectively group similar LOINC codes via interoperability. This method is likely appropriate for many use cases; however, the information contained in the method field of the individual LOINC codes could be important for research questions requiring granular laboratory test information.

Future Directions

Because we designed our study to include heterogeneous data for model development and we included data with rare occurrences (unlike previously published studies) the model may perform well with addition of more tests. Future model development and evaluation could conceivably be attempted without restricting to common tests. We did not include tests with text-reported results due to the need for normalization. However, Hauser *et al*. recently reported creating a scalable, generalizable tool to standardize laboratory test fields,[69] which could potentially be used in conjunction with our method to comprehensively improve data quality and mapping. Additionally, because our manual validation of instances with discordant labels (original source data and model-predicted LOINC codes disagreed) demonstrated model utility for LOINC code resclassification, a future model implementing an iterative training/adjudication/retraining phase could potentially be even more robust for tolerance of noisy labels. For each data instance, the model-predicted label has an associated probability generated by the fitted random forest model. In future iterations, our model could be modified by incorporating label probabilities (as a measure of estimate confidence) to target the data that requires adjudication. Finally, the methods we describe incorporate features created from raw source data aggregation, and as such, could be implemented as an initial step in the transformation pipeline for common data models. We describe a model that effectively maps laboratory data to LOINC codes. However, the same principles could be applied in future applications to map medication data to a terminology (e.g. RxNorm) or to map clinical notes to the LOINC document ontology (set of LOINC codes that classify the key attributes of clinical documents).

Conclusion

With widespread EHR adoption, multi-site data aggregation and centralization are feasible and increasingly common. To leverage these data sources for research, quality assessments, and public health, data must be represented accurately and consistently across sites. Currently, there is a paucity of truly automated methods to map disparate data sources to standards that facilitate consistent data representation. We present a scalable, automated algorithm that may improve data quality and interoperability, while substantially reducing the manual effort currently required to accurately map data.

BIBLIOGRAPHY

1. Chute CG, Cohn SP, Campbell JR. A framework for comprehensive health terminology systems in the United States: development guidelines, criteria for selection, and public policy implications. ANSI Healthcare Informatics Standards Board Vocabulary Working Group and the Computer-Based Patient Records Institute Working Group on Codes and Structures. *J Am Med Inform Assoc.* 1998;5(6):503-510.
2. Ahmadian L, van Engen-Verheul M, Bakhshi-Raiez F, et al. The role of standardized data and terminological systems in computerized clinical decision support systems: literature review and survey. *Int J Med Inform.* 2011;80(2):81-93.
3. eHealth: standardized terminology: report by the Secretariat. *World Health Organization*: Executive Board; 2006:118.
4. Safran C, Bloomrosen M, Hammond WE, et al. Toward a national framework for the secondary use of health data: an American Medical Informatics Association White Paper. *J Am Med Inform Assoc.* 2007;14(1):1-9.
5. Murdoch TB, Detsky AS. The inevitable application of big data to health care. *JAMA.* 2013;309(13):1351-1352.
6. LOINC®. Indianapolis, IN: Regenstrief Institute, Inc. Logical Observation Identifiers Names and Codes (LOINC®). http://www.loinc.org. Accessed December 11, 2017.
7. Regenstrief Institute Inc. *Logical Observation Identifiers Names and Codes (LOINC®)* https://loinc.org/documentation. Accessed 21 Apr 2016.
8. Abhyankar S, Demner-Fushman D, McDonald CJ. Standardizing clinical laboratory data for secondary use. *Journal of biomedical informatics.* 2012;45(4):642-650.
9. Lin MC, Vreeman DJ, McDonald CJ, et al. Correctness of Voluntary LOINC Mapping for Laboratory Tests in Three Large Institutions. *AMIA Annual Symposium proceedings AMIA Symposium.* 2010;2010:447-451.
10. Baorto DM, Cimino JJ, Parvin CA, et al. Combining laboratory data sets from multiple institutions using the logical observation identifier names and codes (LOINC). *Int J Med Inform.* 1998;51(1):29-37.
11. Lin MC, Vreeman DJ, Huff SM. Investigating the semantic interoperability of laboratory data exchanged using LOINC codes in three large institutions. *AMIA Annual Symposium proceedings AMIA Symposium.* 2011;2011:805-814.
12. FitzHenry F, Resnic F, Robbins S, et al. Creating a Common Data Model for Comparative Effectiveness with the Observational Medical Outcomes Partnership. *Appl Clin Inform.* 2015;6(3):536-547.
13. Hersh WR. Adding value to the electronic health record through secondary use of data for quality assurance, research, and surveillance. *Am J Manag Care.* 2007;13(6 Part 1):277-278.
14. Meystre SM, Lovis C, Burkle T, et al. Clinical Data Reuse or Secondary Use: Current Status and Potential Future Progress. *Yearb Med Inform.* 2017;26(1):38-52.
15. The American Recovery and Reinvestment Act of 2009 (ARRA), Public Law 111-5, 123 Stat 115. 17 Feb 2009.
16. Medicare and Medicaid Programs; Electronic Health Record Incentive Program; Final Rule, 42 CFR, §412, 413, 422 *et al.* (2010).
17. Hospitals Participating in the CMS EHR Incentive Programs. *Office of the National Coordinator for Health Information Technology*: Health IT Quick-Stat #45; 2017.
18. McDonald C, Huff S, Deckard J, et al. LOINC Users' Guide. 2017.

19. Lau LM, Johnson K, Monson K, et al. A method for the automated mapping of laboratory results to LOINC. *Proc AMIA Symp.* 2000:472-476.
20. Fidahussein M, Vreeman DJ. A corpus-based approach for automated LOINC mapping. *J Am Med Inform Assoc.* 2014;21(1):64-72.
21. Sun JY, Sun Y. A system for automated lexical mapping. *J Am Med Inform Assoc.* 2006;13(3):334-343.
22. Khan AN, Griffith SP, Moore C, et al. Standardizing laboratory data by mapping to LOINC. *J Am Med Inform Assoc.* 2006;13(3):353-355.
23. Ali T, Khan I, Simpson W, et al. Incidence and outcomes in acute kidney injury: a comprehensive population-based study. *J Am Soc Nephrol.* 2007;18(4):1292-1298.
24. Mohri M, Rostamizadeh A, Talwalkar A. *Foundations of Machine Learning.* Cambridge, MA: The MIT Press; 2012.
25. Samuel AL. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development.* 1959;3(3):210-229.
26. Graves A, Mohamed Ar, Hinton G. Speech recognition with deep recurrent neural networks. Paper presented at: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing; 26-31 May 2013, 2013.
27. Deng L, Li X. Machine Learning Paradigms for Speech Recognition: An Overview. *IEEE Transactions on Audio, Speech, and Language Processing.* 2013;21(5):1060-1089.
28. Tripathy A, Agrawal A, Rath SK. Classification of sentiment reviews using n-gram machine learning approach. *Expert Systems with Applications.* 2016;57:117-126.
29. Bijalwan V, Kumari P, Pascual J, et al. Machine learning approach for text and document mining. *CoRR.* 2014;abs/1406.1580.
30. Xu K, Ba J, Kiros R, et al. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. 2015.
31. Napolitano G, Marshall A, Hamilton P, et al. Machine learning classification of surgical pathology reports and chunk recognition for information extraction noise reduction. *Artif Intell Med.* 2016;70:77-83.
32. Hassanpour S, Langlotz CP. Information extraction from multi-institutional radiology reports. *Artif Intell Med.* 2016;66:29-39.
33. Meystre SM, Kim Y, Gobbel GT, et al. Congestive heart failure information extraction framework for automated treatment performance measures assessment. *J Am Med Inform Assoc.* 2016;24(e1):e40-e46.
34. Ferretti S, Mirri S, Prandi C, et al. Automatic web content personalization through reinforcement learning. *Journal of Systems and Software.* 2016;121:157-169.
35. Deist TM, Dankers F, Valdes G, et al. Machine learning algorithms for outcome prediction in (chemo)radiotherapy: An empirical comparison of classifiers. *Med Phys.* 2018.
36. De Looze C, Beausang A, Cryan J, et al. Machine learning: a useful radiological adjunct in determination of a newly diagnosed glioma's grade and IDH status. *J Neurooncol.* 2018.
37. Walsh CG, Ribeiro JD, Franklin JC. Predicting suicide attempts in adolescents with longitudinal clinical data and machine learning. *J Child Psychol Psychiatry.* 2018.
38. Kononenko I, Matjaz K. *Machine Learning and Data Mining: Introduction to Principles and Algorithms.* Westergate, Chichester, West Sussex, UK: Horwood Publishing; 2007.
39. Agarwal V, Podchiyska T, Banda JM, et al. Learning statistical models of phenotypes using noisy labeled training data. *J Am Med Inform Assoc.* 2016;23(6):1166-1173.
40. Chiu PH, Hripcsak G. EHR-based phenotyping: Bulk learning and evaluation. *Journal of biomedical informatics.* 2017;70:35-51.

41. Simon HU. General Bounds on the Number of Examples Needed for Learning Probabilistic Concepts. *Journal of Computer and System Sciences.* 1996;52(2):239-254.
42. Aslam JA, Decatur SE. On the sample complexity of noise-tolerant learning. *Information Processing Letters.* 1996;57(4):189-195.
43. Sukhbaatar S, Fergus R. Learning from Noisy Labels with Deep Neural Networks. *arXiv preprint arXiv.* 2014;1406.2080.
44. Rolnick D, Veit A, Belongie S, et al. Deep Learning is Robust to Massive Label Noise. *arXiv preprint arXiv.* 2017;1705.10694.
45. Natarajan N, Dhillon IS, Ravikumar P, et al. Learning with noisy labels. Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1; 2013; Lake Tahoe, Nevada.
46. Melville P, Shah N, Mihalkova L, et al. Experiments on Ensembles with Missing and Noisy Data. 2004; Berlin, Heidelberg.
47. Center VIR. *VIReC Factbook: Corporate Data Warehouse (CDW) Consult 2.1 Domain.* Hines IL: U.S Department of Veterans Affairs, Health Services Research & Developement Service, VA Information Resource Center 2014.
48. Center VIR. *VIReC Resource Guide: VistA.* Hines, IL: US Dept of Veterans Affairs, Health Services Research and Development Service, VA Information Resource Center 2012.
49. Jaro MA. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *J Amer Statist Assoc.* 1989;84(406):414-420.
50. Winkler WE. Comparative analysis of record linkage decision rules. *Proceedings of the Section on Survey Research Methodology*: American Statistical Association; 1990:354-359.
51. Jaro MA. Probabilistic linkage of large public health data files. *Stat Med.* 1995;14(5-7):491-498.
52. Levenshtein VI. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady.* 1966;10(8):707-710.
53. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine Learning in Python. *J Mach Learn Res.* 2011;12:2825-2830.
54. Apache cTAKES. http://ctakes.apache.org/. Accessed 15 May 2017.
55. Lin MC, Vreeman DJ, McDonald CJ, et al. Auditing consistency and usefulness of LOINC use among three large institutions - using version spaces for grouping LOINC codes. *Journal of biomedical informatics.* 2012;45(4):658-666.
56. LOINC® Groups. Indianapolis, IN: Regenstrief Institute, Inc. https://loinc.org/groups/. Accessed December 11, 2017.
57. National Library of Medicine; Unified Medical Languague System (UMLS®) REST API Technical Documentation. https://documentation.uts.nlm.nih.gov/rest/home.html. Accessed December 11, 2017.
58. Tibshirani R. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society Series B (Methodological).* 1996;58(1):267-288.
59. Hoerl AE, Kennard RW. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics.* 1970;12(1):55-67.
60. Zou H, Hastie T. Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society Series B (Statistical Methodology).* 2005;67(2):301-320.
61. Breiman L. Random Forests. *Machine Learning.* 2001;45(1):5-32.
62. Bergstra J, Yamins, D., Cox, D. D. Making a Science of Model Search: Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. Paper presented at: 30th International Conference on Machine Learning (ICML 2013) 2013; Atlanta, GA.
63. Chinchor N. MUC-4 evaluation metrics. Proceedings of the 4th conference on Message understanding; 1992; McLean, Virginia.

64. Fisher RA. *The design of experiments.* Oxford, England: Oliver & Boyd; 1935.

65. Student. The Probable Error of a Mean. *Biometrika.* 1908;6(1):1-25.

66. van der Loo M. The stringdist package for approximate string matching. *The R Journal.* 2014;6(1):111-122.

67. R Core Team. R: A Language and Environment for Statistical Computing. *R Foundation for Statistical Computing* 2015; www.R-project.org.

68. rpy2. https://pypi.org/project/rpy2/.

69. Hauser RG, Quine DB, Ryder A. LabRS: A Rosetta stone for retrospective standardization of clinical laboratory test results. *J Am Med Inform Assoc.* 2018;25(2):121-126.