December 2021

Predicting Occurrence of the Term Sarcopenia with Semi-Supervised Machine Learning

Kevin Flasch
University of Wisconsin-Milwaukee

PREDICTING OCCURRENCE OF THE TERM SARCOPENIA WITH SEMI-SUPERVISED MACHINE LEARNING

by

Kevin Flasch

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in Computer Science

at

The University of Wisconsin-Milwaukee

December 2021

ABSTRACT

PREDICTING OCCURRENCE OF THE TERM SARCOPENIA WITH SEMI-SUPERVISED MACHINE LEARNING

by

Kevin Flasch

The University of Wisconsin-Milwaukee, 2021
Under the Supervision of Professor Susan McRoy

Sarcopenia is a medical condition that involves loss of muscle mass. It has been difficult to define and only recently assigned an official medical code, leading to many medical records lacking a coded diagnosis although the clinical note text may discuss it or symptoms of it. This thesis investigates the application of machine learning and natural language processing to analyze clinical note text to see how well the term 'sarcopenia' can be predicted in clinical note text from records concerning the condition.

A variety of machine learning models combined with different features and text processing are tested against training data that mentions the term and test data that is coded for the condition from small datasets from the Medical College of Wisconsin. This research showed that no tested configurations performed exceptionally well, nor combinations of features, based on the F_1 score. Still, some models did show promise, especially those classifying with a support vector machine, as well as other classifiers such as decision trees, gradient boosting and logistic regression. Based on this initial research, while some of the ideas and approaches here did not perform great on the data studied, they provide many some insight and paths forward to extend them and apply them on larger and more precise datasets.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I want to sincerely thank my advisor, Dr. Susan McRoy, for her guidance and continual support throughout this project and during my time at UWM. Her help, patience, and encouragement have been invaluable. I also want to extend my thanks to my committee members, Dr. Ethan Munson and Dr. Jake Luo.

I would like to thank Kristen Osinski, Bradley W. Taylor, George Kowalksi, and Dr. Angela K. Beckert from the Medical College of Wisconsin for their assistance in gathering the data used. I also want to thank Ling Tong for assistance with rating data.

Lastly, I want to express my deep gratitude to my friends and family for all of their support along the way. I especially want to thank my friend Jacob Eisen for all of his support and encouragement to resume my studies, and my parents, Andrea and Bryan Buford, for their unwavering love, support and belief in me.

Chapter 1

Introduction

1.1 About Sarcopenia

Sarcopenia is a condition that entails a progressive and significant loss of skeletal muscle mass and is most commonly related to both aging and immobility. This accelerated decrease is associated with and can lead to frailty, fractures, falls, physical disability, and death. While it is most commonly associated with aging, other influences include genetic and lifestyle (such as exercise, nutrition) factors [1].

It is a condition that has been difficult to find a common definition for, and so also to diagnose well. The International Classification of Diseases (ICD) is a globally used classification of diseases maintained by the World Health Organization that is used for diagnosis (among other clinical tasks) [2]. Sarcopenia was only formally recognized and assigned a specific ICD-10 (the 10th revision of the ICD) code in 2016 [3]. It remains relatively unknown to many clinicians still.

Due to its relatively new status as an actual coded disorder, patients may present with symptoms but a clinician may not know to look for it or to use the appropriate diagnostic tools to look for it specifically. Clinical notes written up for a patient that suffers from sarcopenia may often still not include any specific coding for it, or have other inconsistent

annotations that would make discovery of the condition difficult. Aspects such as these make it difficult to perform retrospective studies of sarcopenia from electronic health records (EHR), to determine how common it occurs, what comorbidities (co-occurring conditions) are most present, or what interventions have been the most successful for patients.

1.2 Motivations and Objectives

How then might this problem of discovery and diagnosis be addressed? Automated methods of natural language processing (NLP) and machine learning (ML) applied to such clinical notes may help. Natural language processing and machine learning provide many tools to analyze and process text. With the multitude of advances in these fields, there is an opportunity to work with text in many new ways. Clinical notes are freehand notes written by clinicians (such as doctors and physicians) describing the status of a patient. Most health care providers typically describe a patient's status this way, while structured data is then added to a clinical record by medical record specialists [4] who read the notes by hand to add such things as ICD codes.

There is still plenty not known on the best ways to analyze and use clinical notes in an automated way with tools such as these. There are many factors that impact how well this can be done, both from the perspective of the tools and algorithms used, the data in question and the general approaches used.

One approach that makes use of these tools and data is to attempt to predict the presence of the term 'sarcopenia' in clinical notes that do not explicitly mention it. Attempting and analyzing this approach is the direction I have chosen to investigate. In this thesis, I look specifically to find answers to the question of what configuration of semi-supervised machine learning can best predict the occurrence of the term sarcopenia in regions of text.

I will compare a variety of machine learning algorithms as implemented in a widely used software library for classification, and compare the use of different features and text

processing based on a dataset of clinical notes relating to sarcopenia. Additionally, this analysis will be done with the constraints of a small dataset without accompanying structured data.

Chapter 2

Background and Related Work

2.1 Data

The data used in this study is comprised of two main sets. One set is to be used as training data and the other as test data. The training and test data are anonymized clinical records with text and minimal structured data provided by the Medical College of Wisconsin (MCW).

The training data is a set of clinical notes where each note has at least one mention of the term 'sarcopenia'. This data is broadly categorized by note type according to MCW, according to how the data was entered in their medical record system. The original query to obtain this data from MCW searched clinical notes for the most occurrences of the term 'sarcopenia', which found 2702 notes across 1416 unique patients across 13 different note types. A sample of this data was then extracted to include the five most common note types: Progress Notes, Consults, H&P, and Discharge Summary (excluding the note type Telephone Encounter) and to roughly 10 notes per note type. A de-identified version of this sample dataset was then provided to us. The size and de-identification of the dataset is necessary as the original data is protected by HIPAA [5].

The test data is a set of clinical notes that do not include the term 'sarcopenia' at all but are all ICD coded positively for sarcopenia according to MCW. The note types included are

the same as those of the training data. This data was de-identified as well and then provided to us. There is no overlap of records between the training and test datasets.

2.1.1 Data Constraints

The datasets are small. The training dataset consists of 40 notes and the test dataset consists of 50 notes after the cleanup and preprocessing. This small set of notes was easier to obtain and have anonymized due to difficulties obtaining large amounts of actual medical data of real patients. Each note does have a large amount of text so there is sufficient data at the sentence level to explore classification methods for each note.

However, such a small set of notes may still impact how well models can be trained. It is also restrictive in that the notes are sourced from the same institution.

2.1.2 Oversampling

Oversampling is one possible way to address the shortcomings of the above data constraints. In this case, oversampling refers to duplicating the positively coded samples in the training dataset a set number of times. Approaches like these (and other more complex dataset manipulations) can help improve performance with small datasets, but they can also easily lead to bad models which overemphasize noise in the data instead of features that are more important (e.g., overfitting).

2.2 Machine Learning Algorithms

Nine different machine learning algorithms for classification were used in this research. They are each briefly described below.

2.2.1 Decision Tree

A decision tree classifier creates a model by making a series of decision rules based on the information gain of combinations of features to arrive at a prediction. Information gain [6] is a measure to determine which features are most important based on the reduction of entropy in decisions.

One nice property of decision trees is that the decision of the classifier at each step can be easy to understand and shown in a visual manner to explain the predictions. That is, a decision tree will tell us not only which feature is associated with a classification, but also what value of each feature was important for that classification. However, decision trees can also easily overfit by creating too complex of trees, or be too sensitive to small variations in the data.

2.2.2 Random Forest

Random forest classification is an ensemble method (a method that uses multiple ML algorithms together) that uses multiple, randomized decision trees over subsets of the data to attempt to shore up weaknesses in regular decision tree classification. Due to this approach, the ability to easily interpret a random forest classifier is mostly lost compared to a single decision tree.

2.2.3 Support Vector Machines (SVM)

Support Vector Machines are a versatile and widely effective machine learning method. They operate by creating hyperplanes for classification across a high or infinite dimensional space based on features and attempt to choose the best hyperplane. The best hyperplane would be where the samples from each side of the plane (the binary classification) are the maximum distance from each other. These samples are referred to as the support vectors.

SVMs tend to generalize very well and so are a good method to try in many applications.

The SVM in this thesis is referred to by SVC, and is a C-Support Vector Classifier (not to be confused with Support Vector Clustering), where C refers to an optimization parameter.

2.2.4 Naïve Bayes

Naïve Bayes classification is a method of probabilistic inference based on Bayes' theorem. The naïve aspect of the algorithm is based on the assumption of conditional independence of every feature. There are variations of the naïve Bayes algorithm. The one used in this research is called multinomial naïve Bayes, where the probabilities are a multinomial distribution, which suits features based on count well, such as word counts in text.

This assumption of conditional independence makes determining probabilities of features occurring given a specific prediction much simpler than if not. These assumptions still lead to classifiers that can perform well in many tasks, including text classification where it has been used often for tasks like spam detection.

2.2.5 Logistic Regression

Logistic Regression is an algorithm used for classification that uses a logistic function as a threshold for linear classification. The probability of a certain class being predicted is based on converting the log-odds (linearly combined features with weights) to a probability which can then be mapped to a binary value with a standard logistic function.

2.2.6 Gradient Boosting

Gradient Boosting is an ensemble learning method that attempts to build a stronger model by optimizing a loss function across weaker models and then adds them together, minimizing the loss function. These weaker models are typically decision trees and so this is sometimes referred to as Gradient Tree Boosting.

2.2.7 k-nearest Neighbors

The k-nearest neighbors (k-NN) classification method is a common form of instance-based (lazy) learning. It makes predictions based on a number (k) of training samples closest in distance to the sample it is predicting. This distance is typically the Euclidean distance from one sample to another across all of its features.

2.2.8 Perceptron and Multi-layer Perceptron

Both Perceptrons and Multi-Layer Perceptrons were used in this research. Perceptrons are a linear classifier that are considered the simplest form of a feedforward (non-cyclic) neural network. Weighted inputs (features) are summed and run through a threshold function for classification. A multilayer perceptron uses multiple layers (referred to as 'hidden' layers) and their activation functions with backpropagation to make predictions. Multi-layer neural networks such as these are the basis of what is known as 'deep learning'.

2.3 Features Used in Text Classification

The features and concepts used to generate features used in this research for classification are described below.

2.3.1 Word Vectorization

Word vectorization is the main technique used in this research to represent the text as features. Typically, a ML algorithm expects fixed-size numerical representations as input for features. Word vectorization converts text into a vector of word counts, so each word across the entire data has a value in each sample, being 0 if it does not occur, or the number of times it occurs in this sample. This is typically referred to as the "bag-of-words" representation, as it is a count of all words with no respect to the order they are in [7].

2.3.2 Named Entity Recognition (NER)

Named entity recognition is a type of information extraction in natural language processing. It involves analyzing text and tagging mentions of "named entities" to a specific category [7]. The categories will vary by the method of NER used, but an example could be matching the word Wisconsin to the category 'Location', or matching malaria to 'Disease'.

Models built for NER are often oriented towards a specific topic or field. NER models specifically for biomedical text are used in this research to transform the clinical text into more general categories.

2.3.3 Lexical Categorization With Empath

Empath [8] is a novel tool that can analyze text across a topic (or 'lexical category') and also generate lexical categories based on words to be used in such an analysis. I used this tool to generate a handful of categories based on a few relevant words and concepts pertaining to sarcopenia. Each chunk can then be analyzed to see if it belongs to any one of these possible categories. It has been used in at least one clinical context [9] to identify topics patients and educators spoke about.

Empath provides a nice way to test a broader categorization of words similar to LIWC (Linguistic Inquiry and Word Count, a program which can also find categories for words, but is not free) [10]. One downside is that Empath features a limited set of models and the categorization creation uses an online backend that does not appear to have its implementation published.

2.3.4 Anatomy (MeSH) Terms

MeSH (Medical Subject Headings) is a specific, controlled vocabulary created by the U.S. National Library of Medicine [11]. It is used in many medical-related contexts to standardize references to medical concepts. It was used in this research to identify and match specific

anatomical terms without introducing the bias of a hand-curated list. The MeSH category for the Musculoskeletal System (A02) [12] was used.

2.3.5 Text Length

A simple feature that can sometimes lead to interesting results is the length of text. When tokenizing the clinical note text by sentence, the chunks will often be of significantly different lengths.

2.4 Libraries and Workbenches for Machine Learning

The software developed for this research was built around the Python machine learning toolkit scikit-learn [13] (also known as sklearn), with initial testing and development being done with the machine learning toolkit Weka.

2.4.1 sklearn and Weka

The Weka Explorer provided an initial easy testbed for experimenting. These experiments were then transitioned to code developed around the library python-weka-wrapper3 for easier repeatability and configuration. As I began integrating more libraries and toolkits with my experiments, and looking at options for other classifiers, I began to take a deeper look at alternatives to Weka, specifically sklearn.

Weka is a solid application and workbench for machine learning tasks involving common ML algorithms. It is easy to run experiments without any code, but can also be integrated as a Java library. However, a lot of modern tooling relating to machine learning is based around Python, as well as NLP libraries. Integrating with these is much easier with a toolkit native to Python, like sklearn. sklearn seems to have wider community support and use for more novel tasks, a good ecosystem of other libraries that work directly with it, and reportedly better performance and memory management (something Weka can struggle with).

All final experimentation code was transitioned to using sklearn.

2.4.2 Additional Libraries

A handful of other libraries and toolkits were utilized in this thesis. Pandas, numpy and liac-arff (allowing easy use of arff files outside of Weka) were used directly with sklearn for managing data. The Natural Language Toolkit (NLTK) was used for various NLP tasks such as sentence tokenization. Empath was used for generating linguistic categories. spaCy and scispaCy were used for Named Entity Recognition. The Wordcloud and matplotlib Python libraries were used for image generation. sklearn-crfsuite was also used for experimentation with Conditional Random Fields, a classifier that did not end up being used.

2.5 Performance Measures

The main performance measures used in this study are as follows.

2.5.1 Precision

Precision is a measure of the proportion of relevant samples among all positive samples. Precision in this context is also called positive predictive value (PPV).

$$Precision = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2.1}$$

2.5.2 Recall

Recall is a measure of what proportion of relevant samples have been found. Recall in this context is also called sensitivity.

$$Recall = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{2.2}$$

2.5.3 F-score

F-score, also called F-measure, is a measure of accuracy calculated from precision and recall. While it may be weighted, it is commonly used (and is used here) as F_1: a balanced F-score, or harmonic mean of precision and recall. It ranges from 0 to 1, with 1 being perfect precision and recall.

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.3}$$

F_1 is the primary metric used in evaluation of the experiments done. It was chosen for similar reasons as in some of the related work. It is a good, balanced measure of both precision in recall, both measures to help determine how well a model can predict positive instances (in this case, occurrences of 'sarcopenia').

2.5.4 Other Measures

A few other measures are referenced but not directly used. They are briefly defined here.

1. Accuracy

$$Accuracy = \frac{\text{True Positives} + \text{True Negatives}}{\text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}} \tag{2.4}$$

2. Negative Predictive Value (NPV)

$$NPV = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Negatives}} \tag{2.5}$$

3. Specificity

$$Specificity = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}} \tag{2.6}$$

4. AUC

   AUC is the area under the ROC (receiver operating characteristic) curve, where the ROC curve is created by plotting the true positive rate against the false positive rate. The AUC is used as a summarization of this curve to give an idea about a model's performance.

2.6 Related Work

There have been many attempts to use both natural language processing and machine learning to analyze the notes found in clinical records. Some of the most relevant and recent ones are discussed here.

A relatively recent systematic review from 2019 [14] looked over many articles on the topic of analyzing clinical notes with NLP for chronic diseases. They carried out a search across several databases that publish articles on these topics, such as Scopus, Web of Science / MEDLINE, PubMed, and the ACM Digital Library across terms related to clinical notes, NLP, and chronic diseases from January 1, 2007 to February 6, 2018. Their overview showed the rapid increasing use of machine learning in such applications, with rule-based methods still used as well. The machine learning employed is generally shallower classifiers (especially SVM and Naïve Bayes) as opposed to deep learning, which appeared a bit more uncommon in this particular area, despite the apparent potentials for it [15]. Much of the reviewed work focuses on classification of disease phenotypes, or traits. They identify a lack of more complicated information extraction approaches and much use of structured data. However, as shown in one study [16], there are many cases of much more clinical note information available to be analyzed than structured data. They also identify the issue of much research relying on small datasets, likely due to a general lack of access to larger amounts of clinical data.

Another somewhat recent and relevant review [17] concerns information extraction from

clinical records. Their review is based on a search of articles from relevant databases such as Ovid MEDLINE, Ovid EMBASE, Scopus, Web of Science, and ACM Digital Library from January 1, 2009 to September 6, 2016. They found that most data in electronic health records is free-text [18], as opposed to structured data. Also, much analysis is still rule-based, such as research using regular expressions for matching terms for peripheral arterial disease [19], with machine learning methods typically being used for predictions, estimations, and finding associations as opposed to uses for information extraction. They analyzed the most frequently used clinical information extraction tools in studies, with the most common ones being cTAKES, MetaMap, and MedLEE. For machine learning, SVM was shown to be the most widely used ML method among their results. They also found a lack of deep learning approaches compared with non-clinical NLP research. This review also indicated issues relating to limited access of health records to researchers [20], which also in turn affects generalizing of clinical information extraction. They identify possible solutions that include cross-disciplinary training of NLP researchers to increase understanding across biomedical domains, and adoption of standards to help collaboration and access of data.

A study by Weng et al. [21] shows good performance in utilizing supervised deep learning and shallower learning algorithms in binary classification of clinical notes according to medical subdomains (e.g., cardiology, neurology, etc). They utilized two datasets: an iDASH (a publicly available anonymized repository) dataset of 431 diverse clinical notes and a dataset of 542,744 clinical notes based on specialist visits from the Massachusetts General Hospital. Their study utilized only the unstructured clinical notes, and built features based on that: a bag-of-words text representation, and different groupings based on UMLS (Unified Medical Language System) concepts extracted with the tool cTAKES (Apache clinical Text Analysis and Knowledge Extraction System). The classifiers they utilized were multinomial naïve Bayes, logistic regression, SVMs with a linear kernel and with stochastic gradient descent, random forest, adaptive boosting, and two deep learning classifiers, a convolutional neural network and convolutional recurrent neural network. They evaluated the performance of

the classifiers with balanced accuracy, precision, recall, F_1 score, and AUC across various combinations of those features. The iDASH dataset performed best among the shallow classifiers with a linear kernel SVM (F_1 scores of 0.927 to 0.932 and AUC of 0.955 to 0.957 depending on features), and the MGH dataset performed best with a linear kernel SVM as well, with logistic regression performing well too (F_1 scores of 0.915 to 0.934 and AUC of 0.953 to 0.964). The deep learning classifiers were considered to have performed better when evaluating by both AUC and F_1, with slightly lower F_1 scores but higher AUC scores than the shallower classifiers.

Venkataraman et al. [22] documents a very recent attempt to build a system to automate assignment of ICD codes to clinical records (both human and veterinary records in this case) via deep learning. The study utilized datasets of 89,591 records from the veterinary teaching hospital at Colorado State University, and 52,722 records from the MIMIC-III database, a publicly available dataset from the Beth Israel Deaconess Medical Center of Boston, Massachusetts. This was also a case of supervised learning, with the datasets coded already (although in the case of the veterinary records, the provided codes had to be translated to ICD codes). Only unstructured clinical note text was used (as well as the codings) with no other structured information from the records. The study produced baseline results with decision tree and random forest classifiers, using tf-idf representation of words as their features. The deep learning classifier used was a long short-term memory (LSTM) recurrent neural network (RNN), where words were represented densely with word embeddings, using GloVe (Global Vectors for Word Representation) [23]. They also investigated using MetaMapLite [24] as a text transformation tool to help consolidate medical information, though did not find much gain from its use (though it was indicated it or similar approaches may be of more use in the future with more work). The testing done involved different iterations of their datasets as either the training data or test data or combined versions of the datasets as the same. They found some promising results with their deep learning approach, with scores generally slightly higher than their shallow classifier baselines in most cases. This study

evaluated performance with precision, recall, and F_1 score, primarily ranking it on an average weighted macro F_1 score (this averaging due to it being a multi-label problem). The F_1 scores of the LSTM classifier ranged from 0.66 to 0.91, with the best performance using the CSU dataset for both training and test and with no difference observed using MetaMapLite. Using the combined datasets for both MIMIC and CSU resulted in the random forest classifier having the best F_1 score (0.086) when not using MetaMapLite, but the LSTM scored with an F_1 of 0.90 when using MetaMapLite in that instance.

Another study by Wang et al. [25] describes an approach of clinical notes classification on smoking status and hip fractures using weak supervision. They utilized datasets of two case studies from Mayo Clinic regarding smoking status (32,336 records) and hip fracture classification (22,969 records), and one public dataset from i2b2 of a 2006 smoking status classification study (389 records). The weak supervision consisted of taking random samples of test data from the Mayo Clinic datasets and having them coded by a medical expert, and then running a rule-based NLP algorithm to label the remaining training data. They extracted the coding for the i2b2 themselves with a rule-based system as well. These automatic codings were all performed on the clinical note text alone, with the rule-based method of coding based on pattern searches of relevant words and phrases on the text. The classifiers they tested along with the rule-based system were SVM, Random Forest (RF), Multilayer Perceptron Neural Networks (MLPNN), and Convolutional Neural Networks (CNN). Words were represented as word embeddings, which required some conversion steps to features for the shallower classifiers. They compared this conversion in SVM and RF with tf-idf representation and topic modeling. Their use of word embeddings in SVM and RF classifiers showed better results than using tf-idf and topic modeling on both datasets, with SVM having an F_1 score of 0.80 vs 0.69 (tf-idf) and 0.73 (topic modeling) on the Mayo Clinic data, SVM having an F_1 score of 0.95 vs 0.85 (tf-idf) and 0.91 (topic modeling) and RF performing mostly similarly. For the comparisons with the deep learning methods, they found the best performance with a CNN on the Mayo Clinic smoking data (F_1 of 0.92) and the Mayo

Clinic hip fracture data (F_1 of 0.97), which they indicated as statistically significantly better than the other methods. For the i2b2 dataset, the CNN performed worst (F_1 of 0.77) and the best performer was the rule-based NLP method, with an F_1 of 0.88, with the shallower methods not far behind. Precision and recall were also measured alongside F_1 and had the same rankings. They indicate the performance difference is likely due to the size of the i2b2 dataset compared to the others, as the CNN is more resistant to smaller data sizes to build up an accurate model. Still, the study shows an algorithmic process like this, with a large and well-formed dataset, could greatly help reduce manual human labeling of training data.

Afzal et al. [26] describes using NLP for finding cases of peripheral arterial disease (PAD) in clinical notes. This study procured data from the Mayo Clinic's clinical data warehouse, using a training dataset of 300,364 clinical notes across 935 patients and a test dataset of 212,047 clinical notes across 634 patients. While not utilizing machine learning, their approach uses text processing to find related concepts and rule-based methods for classifying patients. Their rule-based method operated on the text alone with no use of additional structured information from the clinical records. It utilized MedTagger [27], a tool for identifying medical concepts, which it found and then mapped to categories relevant to PAD. Their process also used keywords in text to identify positive, negative and possible status of concepts. They evaluated performance with accuracy, positive predictive value (PPV, precision), sensitivity (recall), negative predictive value (NPV), and specificity. Their system was compared against classification of data merely by billing codes (i.e., ICD codes) or a combination of billing codes and procedural codes (structured data indicating patient procedure). Their NLP algorithm performed with accuracy greater than analysis by codes alone on the test data (91.8 vs 81.1 and 83.0), although it was weaker in sensitivity than the systems using billing and procedural codes (91.2 vs 97.0) and NPV (90.7 vs 95.2). Reasons for these lower scores were indicated as false positives from notes where it was suspected the patient had PAD by the clinician, but later tests ruled it out, and cases where their NLP algorithm was unable to differentiate the experiencer of a disease.

Specifically related to sarcopenia, there has been some work related to analyzing clinical records for it. One such paper [28] looked at clinical records to attempt to build a phenotype (in this case, a list of characteristics of a person that appear to be positively associated with specific conditions) for sarcopenia, frailty and cachexia (a disorder that can cause muscle wasting). They analyzed records from the Indiana Network for Patient Care between 2016 and 2017. All records from eligible patients in the system's database were examined (18 years of age and older, and having encounters and clinical notes within the Indiana University Health System and Eskenazi Health Systems in the given time range). They generated the phenotype based on ICD-9 and ICD-10 codes for frailty and cachexia, and the ICD-10 code for sarcopenia, and by searching the clinical notes for the terms (and variants of) sarcopenia, frailty and cachexia, using in-house NLP software (nDepth). This NLP-aided search attempted to work with misspellings, grammar variants and negations. They then used their computed phenotype to detect 10,288 records in the database between 2016 and 2017. These were reviewed by two clinician investigators, who found 9594 (93.3%) were positive cases where a clinician identifying the patient as having one of the study's conditions. The other 694 (6.7%) records indicated a negation of one of the conditions or its presence in someone other than the patient. They found most cases were detected by text terms without ICD code at all (86.4%). All cases detected via ICD codes also has supportive text terms. In particular, sarcopenia was only detected by ICD code in 10 patients while text terms found it in 310. These results were compared to a set of controls who matched by birth year, sex and race but had no related ICD coding or text terms which found some difference in clinical variables (suggesting more search criteria for these conditions). Overall, this study shows good results in detecting these conditions via clinical note text where they are not specifically coded, but there are distinct references to them. The records of patients detected by such a phenotype can be useful in identifying characteristics of these conditions to look for in clinical notes where the terms themselves might not exist.

Chapter 3

Methodology

This section describes the methodology used for this study. It was a process of data preparation and processing, feature creation and selection, classifier model comparison and test data predictions.

3.1 Data Acquisition and Processing

The data used was obtained from MCW as described earlier in the Data section of the Background as two main datasets in CSV format, with the training dataset comprised of notes mentioning the term 'sarcopenia', and test set comprised of positively ICD coded notes for sarcopenia but not including the term. Aside from the clinical note text, they contain note type, note id, and in the case of the training data, patient id and encounter id, all anonymized by MCW. Training (via the CITI Program [29]) was required and undergone by myself to understand how to properly handle and make use of sensitive personal medical information.

3.1.1 Data Cleanup and Preprocessing

The training data provided required some additional cleanup before processing. Four notes did not actually contain the term 'sarcopenia' at all and were discarded. Four other notes were duplicates (with the same id and text) and were removed. A few other notes (2 from one patient and encounter, and 5 from another patient and encounter) were distinct but very similar to each other and appeared to be small updates to each patient's encounter. In these cases, the latest note was used for the encounter and the rest discarded.

The resulting training dataset are distinct notes that all contain at least one mention of the term 'sarcopenia'. The test dataset provided contained no duplicate note ids (and so no exact duplicate notes). A few notes appear to be updates to previous notes similar to the examples in the test data. These were retained, as the test data is not used to create the model and updates may include or remove text relevant to testing classification. Thus, the test data required no pruning. The dataset CSV files were then converted to UTF-8 to reduce friction during analysis and experiments.

Finally, the notes in the datasets are converted into samples. As the intention is to predict regions of notes that might concern the concept 'sarcopenia', the notes were segmented into various chunk sizes to determine which sized chunk performed the best. This chunking was primarily done with sentence tokenization (via NLTK) to separate each note into n-sentence chunks.

To attempt to capture context surrounding the occurrence of the word 'sarcopenia' in the training data, this tokenization centered around each sentence that contained the term. Chunk sizes of 1, 3, 5, 7 and 9 sentences were all tested, with an even number of sentences (where the chunk size is greater than 1) surrounding each sentence where the term occurred. The rest of the note is tokenized by the same value (which may result in 1 or 2 chunks at the beginning or the end of the note not exactly matching the sentence count). Each chunk containing the term 'sarcopenia' is positively coded, and the term is then removed from the text in the samples to prevent the models from merely training on the word itself.

The test data is tokenized by the same amount, but as the term 'sarcopenia' does not occur in the test data, the chunking does not take into account any specific sentence or location as in the training data and begins from the start of the text.

3.1.2 Test Dataset Coding

The test dataset, as described, is entirely positively ICD coded for sarcopenia. To be able to utilize individual chunks of the test data for comparison, each chunk must be assigned a positive or negative classification. This rating was performed by myself and two other volunteers, who had completed CITI training and were approved under the project's IRB protocol, to provide another measure of performance. The most promising chunk size was chosen to rate, and each person rated each chunk of text as: 0 (does not suggest sarcopenia), 3 (unsure or may suggest sarcopenia), 5 (suggests sarcopenia). There was no attempt to manually resolve differences between each individual's ratings. Instead, the ratings were averaged together.

3.1.3 Oversampling

A simple test using oversampling was also done. Positively coded samples in the training data set were duplicated in different amounts to see the impact on performance.

3.1.4 Note Text Analysis

The note text was analyzed in a few different ways to understand its composition. An average of the characters per note and sentences per note in each dataset was calculated. A "word cloud" visualization was performed to generate a visual representation of the most common tokens (e.g., words) in the datasets. Some extremely common tokens were filtered from this visualization such as "XXXXX" (used all over for anonymization) and "patient", common English stopwords, and numbers.

An analysis based on F_1 scores across chunk sizes (based on sentences) was done on sizes of 1, 3, 5, 7 and 9. These chunk sizes were tested with 10-fold cross-validation across every classifier with no features (aside from text), one of the main features, and all features. The top 50% of results as determined by F_1 score were compared on frequency to determine which chunk sizes to focus on.

3.2 Features

A combination of different features which are described below were used and tested. 10-fold cross-validation was done to compare these different features to see which had promise and which did not.

3.2.1 Clinical Note Text

The main feature used across all training is the clinical note text itself. The text in each sample, which is comprised of an n-sentence chunk, is converted via word vectorization, where it is transformed into a matrix of word counts. In all cases it also converts the text to lowercase beforehand, as well. The tokenization to determine words is based on 2 or more alphanumeric characters with punctuation treated only as a token separator.

1. Named Entity Recognition An attempt was made to utilize named entity recognition to reduce the note text to specific entities. A project that builds biomedical models to be used for named entity recognition and other NLP tasks with the spaCy toolkit called scispaCy was used. This approach was used to see if reducing the note text to mostly recognized biomedical concepts might improve classification.

2. Other Text Processing A few other techniques were also tested to see what impact they had. A conversion of the words vectors into bigrams was tested. Conversion of the word vectors to tf-idf representation was tested. Use of simple stopwords was tested.

3.2.2 Note type

As discussed in the data section, the data was initially retrieved based on the most prevalent occurring note types in the MCW database. This provided a label for each note (and therefore each chunk) which was easily adapted into a feature.

3.2.3 Text length

This length of each chunk was utilized as a feature.

3.2.4 Empath

Categories were generated with Empath based on words related to sarcopenia. Five categories were created with specific seed words shown below. Each category is a feature indicating if a word in that chunk is present or not. Other related seed words were tried but found lacking in the results returned from Empath.

Table 3.1 shows the five categories used as features and the words supplied to Empath to generate those categories.

Table 3.1: Empath Categories

Category	Words Used to Create Category
depleting	depleting
muscskel	muscle, skeletal, musculoskeletal
gaitmobility	gait, mobility
fracture	fracture
frail	frail, frailty

3.2.5 Anatomy (MeSH) terms

A feature generated from a list of anatomy terms pertaining to the musculoskeletal system from the MeSH A02 category was used. For each chunk, this feature indicated if any terms from this list were present.

3.3 Machine Learning Methods

A variety of different learning methods were used and tested, as detailed above in Machine Learning Algorithms. 10-fold cross-validation was used to compare different iterations of classifiers and features.

3.4 Experimental Process

Initial analysis was done with Weka, but the process was transitioned to sklearn with various libraries to better integrate with the large Python ML and NLP ecosystem as described in a previous section. Each classifier listed above was analyzed with 10-fold cross-validation across the different features. Different performance measures were collected, most importantly precision, recall and F_1 score as the main measures used in this study.

These cross-validation results were used to determine the most promising chunk size to be used in the rest of the analysis.

The test dataset was then rated by the most promising chunk from the cross-validation analysis as described above.

Chapter 4

Results and Discussion

The results of a variety of different experiments based on the methodology above are described here.

4.1 Note Text Analysis

Table 4.1 shows information about text in the datasets, detailing the number of notes, average number of sentences per note, and average number of characters per sentence.

Table 4.1: Average Values Per Note Per Dataset

Dataset	# Notes	Avg. Sentences Per Note	Avg. Chars Per Sentence
Training	40	112	84
Test	50	68	82

Figures 4.1 to 4.4 provide word cloud visualizations for the most frequently occurring terms in the notes for all notes in the training data, all notes in the training data for the positive class, all notes in the test data, and all notes in the test data for the positive class, respectively.

Figure 4.1: Training Data Word Cloud for all Note Text

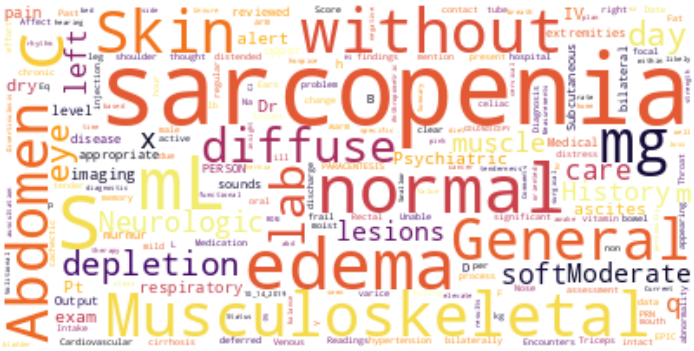

Figure 4.2: Training Data Word Cloud for Sentences Surrounding 'Sarcopenia'

Figure 4.3: Test Data Word Cloud for all Note Text

Figure 4.4: Test Data Word Cloud for 5 Sent. Chunks With Ratings > 0

4.2 10-fold Cross-Validation Results

4.2.1 Initial Weka Results

The results in Table 4.2 are from initial cross-validation testing with Weka. They provide a point of comparison with the results produced by sklearn. While other features and chunk sizes were analyzed, merely the notes as a simple word vector and chunk size of 5 is included here for brevity.

Table 4.2: Weka Cross-Validation Results On 5 Sent. Notes

Classifier	Precision	Recall	F_1
SMO (Support Vector)	0.857	0.462	**0.600**
NaiveBayes	0.316	**0.641**	0.424
J48 (Decision Tree)	0.417	0.256	0.317
IBk (k-Nearest Neighbors)	**1.000**	0.179	0.304
RandomForest	**1.000**	0.051	0.098

4.2.2 Comparison Of Chunk Sizes

Figure 4.5 shows the distribution of the top 50% performing chunks by F_1 score across all classifiers using combinations of only notes as features, each major feature individually with notes, and all features combined.

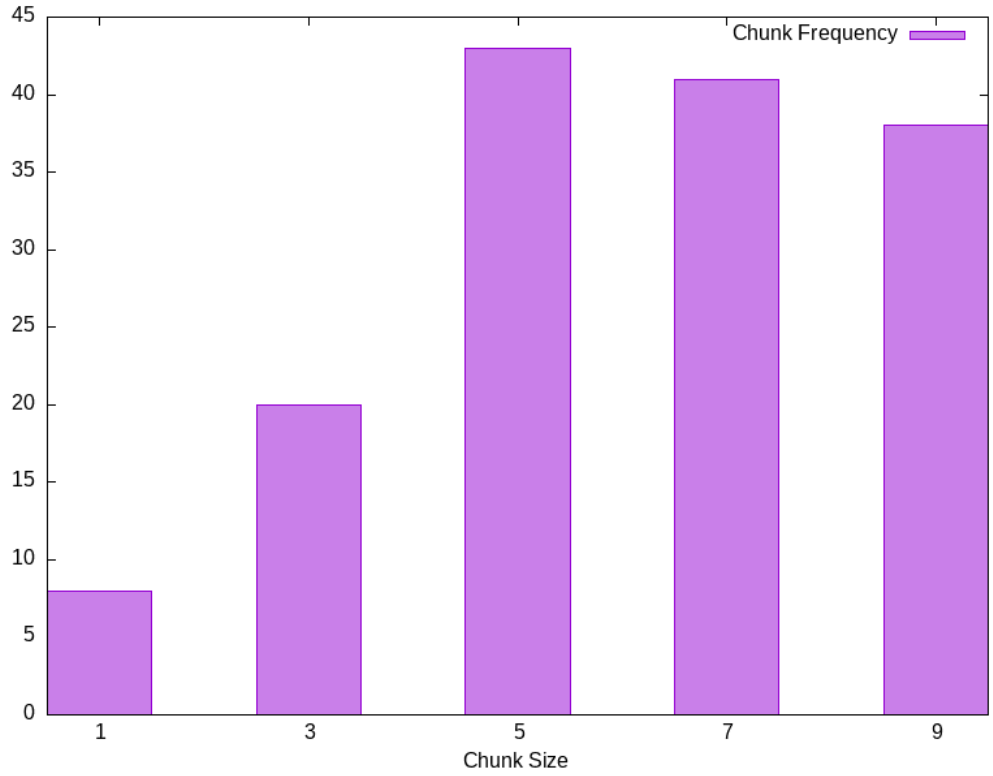

Figure 4.5: Chunk Frequency In Top 50% F_1 Results

Based on the above, the remaining experiments focus on a chunk size of 5 and 7 sentences.

4.2.3 Analysis Based on Note Text

Table 4.3 shows the results of 10-fold cross-validation of the training data on 5 and 7 sentence chunk notes with no other features. Table 4.4 shows the results of 10-fold cross-validation of the training data on 5 and 7 sentence chunk notes transformed into bigrams with no other features. Table 4.5 shows the results of 10-fold cross-validation of the training data on 5 and 7 sentence chunk notes with notes transformed via named entity recognition, using the scispaCy model "en_core_sci_sm", with no other features.

Table 4.3: Cross-Validation Results on 5 and 7 Sent. Notes

Classifier	Chunk	Precision	Recall	F_1
SVC	7	0.840	0.538	**0.656**
SVC	5	0.818	0.462	0.590
GradientBoostingClassifier	7	0.850	0.436	0.576
DecisionTreeClassifier	7	0.594	0.487	0.535
LogisticRegression	7	0.833	0.385	0.526
GradientBoostingClassifier	5	0.789	0.385	0.517
MLPClassifier	5	0.696	0.410	0.516
LogisticRegression	5	0.929	0.333	0.491
DecisionTreeClassifier	5	0.531	0.436	0.479
MLPClassifier	7	0.700	0.359	0.475
Perceptron	5	0.452	0.487	0.469
Perceptron	7	0.425	0.436	0.430
RandomForestClassifier	5	**1.000**	0.231	0.375
KNeighborsClassifier	5	0.462	0.308	0.369
KNeighborsClassifier	7	0.458	0.282	0.349
MultinomialNB	5	0.223	**0.590**	0.324
MultinomialNB	7	0.208	0.513	0.296
RandomForestClassifier	7	**1.000**	0.154	0.267

Table 4.4: Cross-Validation Results on 5 and 7 Sent. Notes with Bigrams

Classifier	Chunk	Precision	Recall	F_1
GradientBoostingClassifier	7	0.905	0.487	**0.633**
DecisionTreeClassifier	7	0.750	0.538	0.627
DecisionTreeClassifier	5	0.864	0.487	0.623
GradientBoostingClassifier	5	0.833	0.385	0.526
MLPClassifier	7	0.667	0.410	0.508
SVC	7	**1.000**	0.256	0.408
SVC	5	0.909	0.256	0.400
MLPClassifier	5	0.647	0.282	0.393
Perceptron	7	0.328	0.487	0.392
Perceptron	5	0.258	0.436	0.324
RandomForestClassifier	5	**1.000**	0.128	0.227
RandomForestClassifier	7	**1.000**	0.103	0.186
LogisticRegression	7	**1.000**	0.103	0.186
MultinomialNB	7	0.086	**0.897**	0.158
LogisticRegression	5	**1.000**	0.077	0.143
MultinomialNB	5	0.060	0.821	0.113
KNeighborsClassifier	5	0.333	0.051	0.089
KNeighborsClassifier	7	**1.000**	0.026	0.050

Table 4.5: Cross-Validation Results on 5 and 7 Sent. Notes with NER en_core_sci_sm

Classifier	Chunk	Precision	Recall	F_1
DecisionTreeClassifier	7	0.719	0.590	**0.648**
SVC	5	0.714	0.513	0.597
SVC	7	0.783	0.462	0.581
MLPClassifier	7	0.704	0.487	0.576
LogisticRegression	7	0.941	0.410	0.571
MLPClassifier	5	0.655	0.487	0.559
LogisticRegression	5	0.833	0.385	0.526
GradientBoostingClassifier	7	0.714	0.385	0.500
Perceptron	7	0.409	0.462	0.434
GradientBoostingClassifier	5	0.706	0.308	0.429
Perceptron	5	0.364	0.513	0.426
KNeighborsClassifier	5	0.786	0.282	0.415
RandomForestClassifier	7	**1.000**	0.205	0.340
DecisionTreeClassifier	5	0.342	0.333	0.338
MultinomialNB	7	0.205	0.590	0.305
RandomForestClassifier	5	**1.000**	0.179	0.304
MultinomialNB	5	0.192	**0.615**	0.293
KNeighborsClassifier	7	0.700	0.179	0.286

4.2.4 Analysis Based on Note Text and Other Features

Tables 4.6 - 4.9 show the results of 10-fold cross-validation of the training data on 5 and 7 sentence chunk notes, each with one additional major feature (note type, text length, Empath, MeSH anatomy terms). Table 4.10 shows the results of 10-fold cross-validation of the training data on 5 and 7 sentence chunk notes with all main features (note type, text length, Empath, MeSH anatomy terms) used.

Table 4.6: Cross-Validation Results on 5 and 7 Sent. Notes and Note Type

Classifier	Chunk	Precision	Recall	F_1
SVC	7	0.840	0.538	**0.656**
SVC	5	0.826	0.487	0.613
GradientBoostingClassifier	7	0.818	0.462	0.590
LogisticRegression	7	0.833	0.385	0.526
DecisionTreeClassifier	5	0.600	0.462	0.522
Perceptron	7	0.600	0.462	0.522
MLPClassifier	5	0.667	0.410	0.508
Perceptron	5	0.500	0.487	0.494
DecisionTreeClassifier	7	0.567	0.436	0.493
GradientBoostingClassifier	5	0.682	0.385	0.492
LogisticRegression	5	0.929	0.333	0.491
MLPClassifier	7	0.515	0.436	0.472
RandomForestClassifier	7	**1.000**	0.231	0.375
KNeighborsClassifier	5	0.462	0.308	0.369
RandomForestClassifier	5	**1.000**	0.205	0.340
MultinomialNB	5	0.234	**0.564**	0.331
KNeighborsClassifier	7	0.435	0.256	0.323
MultinomialNB	7	0.226	0.487	0.309

Table 4.7: Cross-Validation Results on 5 and 7 Sent. Notes and Text Length

Classifier	Chunk	Precision	Recall	F_1
SVC	7	0.808	0.538	**0.646**
SVC	5	0.818	0.462	0.590
GradientBoostingClassifier	7	0.783	0.462	0.581
GradientBoostingClassifier	5	0.833	0.385	0.526
LogisticRegression	7	0.833	0.385	0.526
DecisionTreeClassifier	7	0.559	0.487	0.521
DecisionTreeClassifier	5	0.500	0.487	0.494
LogisticRegression	5	0.929	0.333	0.491
MLPClassifier	7	0.552	0.410	0.471
Perceptron	7	0.500	0.410	0.451
MLPClassifier	5	0.560	0.359	0.438
Perceptron	5	0.390	0.410	0.400
KNeighborsClassifier	5	0.481	0.333	0.394
KNeighborsClassifier	7	0.458	0.282	0.349
MultinomialNB	5	0.224	**0.564**	0.321
RandomForestClassifier	5	**1.000**	0.179	0.304
RandomForestClassifier	7	**1.000**	0.179	0.304
MultinomialNB	7	0.209	0.487	0.292

Table 4.8: Cross-Validation Results on 5 and 7 Sent. Notes and Empath

Classifier	Chunk	Precision	Recall	F_1
SVC	7	0.840	0.538	**0.656**
SVC	5	0.818	0.462	0.590
GradientBoostingClassifier	7	0.818	0.462	0.590
DecisionTreeClassifier	7	0.633	0.487	0.551
MLPClassifier	7	0.633	0.487	0.551
LogisticRegression	7	0.833	0.385	0.526
GradientBoostingClassifier	5	0.789	0.385	0.517
Perceptron	7	0.513	0.513	0.513
DecisionTreeClassifier	5	0.562	0.462	0.507
LogisticRegression	5	0.929	0.333	0.491
MLPClassifier	5	0.577	0.385	0.462
Perceptron	5	0.400	0.410	0.405
KNeighborsClassifier	5	0.462	0.308	0.369
KNeighborsClassifier	7	0.458	0.282	0.349
RandomForestClassifier	5	**1.000**	0.205	0.340
MultinomialNB	5	0.223	**0.590**	0.324
MultinomialNB	7	0.208	0.513	0.296
RandomForestClassifier	7	**1.000**	0.154	0.267

Table 4.9: Cross-Validation Results on 5 and 7 Sent. Notes and Anatomy Terms

Classifier	Chunk	Precision	Recall	F_1
SVC	7	0.833	0.513	**0.635**
SVC	5	0.818	0.462	0.590
GradientBoostingClassifier	7	0.818	0.462	0.590
GradientBoostingClassifier	5	0.833	0.385	0.526
LogisticRegression	7	0.833	0.385	0.526
MLPClassifier	5	0.630	0.436	0.515
DecisionTreeClassifier	7	0.586	0.436	0.500
Perceptron	7	0.529	0.462	0.493
LogisticRegression	5	0.929	0.333	0.491
DecisionTreeClassifier	5	0.500	0.385	0.435
MLPClassifier	7	0.483	0.359	0.412
Perceptron	5	0.390	0.410	0.400
KNeighborsClassifier	5	0.462	0.308	0.369
KNeighborsClassifier	7	0.458	0.282	0.349
MultinomialNB	5	0.223	**0.590**	0.324
MultinomialNB	7	0.208	0.513	0.296
RandomForestClassifier	5	**1.000**	0.154	0.267
RandomForestClassifier	7	**1.000**	0.154	0.267

Table 4.10: Cross-Validation Results on 5 and 7 Sent. Notes with Note Type, Text Length, Empath and Anatomy Terms

Classifier	Chunk	Precision	Recall	F_1
SVC	7	0.800	0.513	**0.625**
SVC	5	0.826	0.487	0.613
GradientBoostingClassifier	7	0.850	0.436	0.576
Perceptron	5	0.600	0.538	0.568
DecisionTreeClassifier	7	0.655	0.487	0.559
LogisticRegression	7	0.833	0.385	0.526
MLPClassifier	7	0.586	0.436	0.500
GradientBoostingClassifier	5	0.778	0.359	0.491
LogisticRegression	5	0.929	0.333	0.491
DecisionTreeClassifier	5	0.531	0.436	0.479
Perceptron	7	0.625	0.385	0.476
MLPClassifier	5	0.519	0.359	0.424
KNeighborsClassifier	5	0.481	0.333	0.394
KNeighborsClassifier	7	0.458	0.282	0.349
RandomForestClassifier	5	**1.000**	0.205	0.340
MultinomialNB	5	0.242	**0.564**	0.338
MultinomialNB	7	0.235	0.487	0.317
RandomForestClassifier	7	**1.000**	0.179	0.304

4.2.5 Analysis of Oversampling

Table 4.11 shows the results of 10-fold cross-validation of the training data on 5 and 7 sentence chunk notes, with the positive samples oversampled four times.

Table 4.11: Cross-Validation Results on 5 and 7 Sent. Notes With Oversampling of 4

Classifier	Chunk	Precision	Recall	F_1
RandomForestClassifier	5	**1.000**	**1.000**	**1.000**
RandomForestClassifier	7	**1.000**	**1.000**	**1.000**
GradientBoostingClassifier	7	0.985	1.000	0.992
SVC	5	0.980	1.000	0.990
MLPClassifier	5	0.980	1.000	0.990
GradientBoostingClassifier	5	0.970	1.000	0.985
LogisticRegression	5	0.970	1.000	0.985
SVC	7	0.970	1.000	0.985
LogisticRegression	7	0.961	1.000	0.980
MLPClassifier	7	0.942	1.000	0.970
KNeighborsClassifier	5	0.933	1.000	0.965
KNeighborsClassifier	7	0.920	1.000	0.958
DecisionTreeClassifier	7	0.894	1.000	0.944
Perceptron	5	0.878	1.000	0.935
DecisionTreeClassifier	5	0.863	1.000	0.926
Perceptron	7	0.830	1.000	0.907
MultinomialNB	7	0.682	1.000	0.811
MultinomialNB	5	0.603	0.979	0.746

4.3 Test Data Prediction Results

Figure 4.6 shows the distribution of ratings of each chunk of the test set, where the three ratings of each chunk were averaged together.

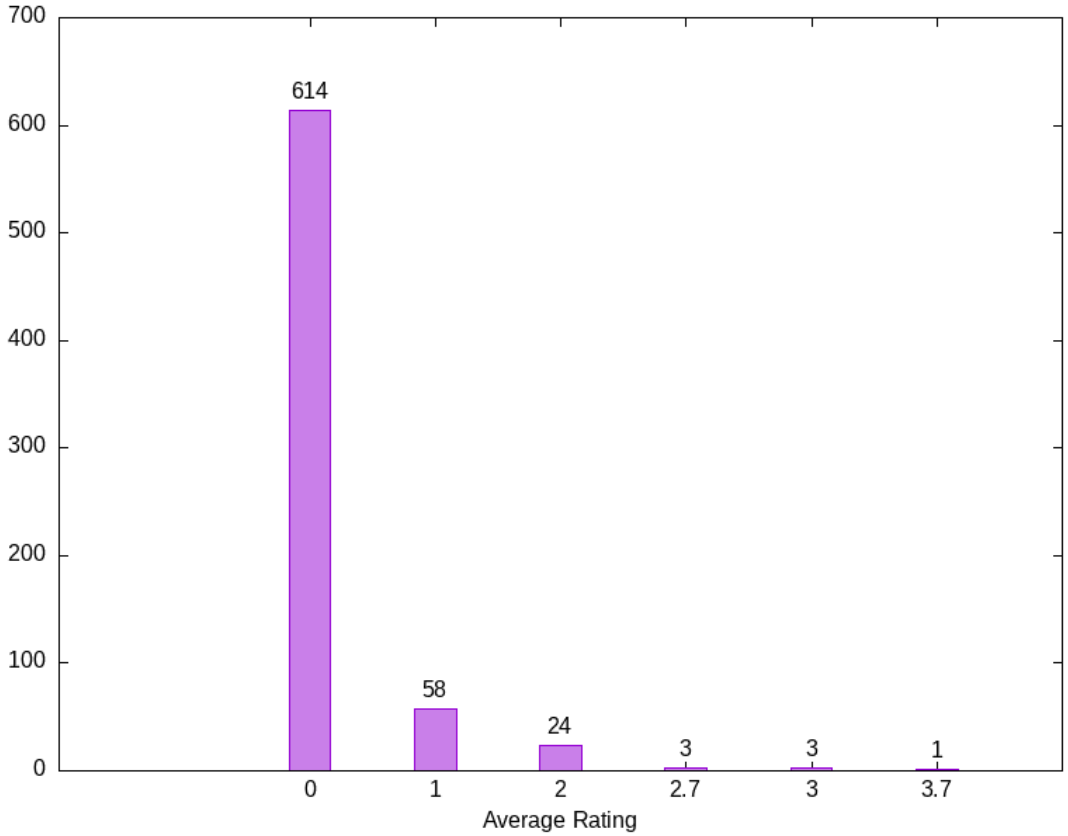

Figure 4.6: Test Data Ratings Distribution

Table 4.12 shows examples of five sentence chunks (truncated for length) of test data based on ratings given.

Table 4.12: Examples of Rated Test 5 Sentence Chunks

Rating	Sentence Chunk
0	Acute interstitial edematous pancreatitis. No peripancreatic fluid collection. 2. Enlarged left hepatic and caudate [XXXXX] , which raises the possibility of underlying liver disease. 3.
0	No distress. HENT : [XXXXX] : Normocephalic and atraumatic. Cardiovascular : Normal rate and regular rhythm. Exam reveals no gallop and no friction rub. Murmur (systolic) [XXXXX] .
3	Perioperative Medicine Progress Note Service : Perioperative Medicine Date of Service : [5_30_2019] Chief Complaint : Mechanical Fall Brief History : 88 y / o female w / PM Hx of dementia , A. fib on Warfarin , CHB s / p dual chamber PM (2012) , HTN , RA , h / o CVA (2017) , HLD , chronic pain , and urinary retention (straigh cath dependent). Pt presented to the ED on [5_26] after a witnessed mechanical fall. Pt reportedly was bending [XXXXX] and lost her balance. She [XXXXX] onto her left side. Did not hit [XXXXX] .
3	2. Multilevel degenerative changes. [5_21] EKG : Paced rhythm , ventricular rate 75 Diagnostic and Therapeutic Plan : 88 y / o female with PM Hx of dementia , A. fib (on Warfarin CHADSVASC 6) , CHB s / p dual chamber PM (2012) , HTN , RA w / cervical instability , h / o CVA (lacunar infarcts on CT in 2017) , HLD , urinary retention requiring straight catheterization , and chronic [XXXXX] pain [2_6] DDD w / chronic narcotic use. Pt had a mechanical fall resulting in left hip comminuted angulated and displaced intertrochanteric femur fracture. Taken to OR on [5_21] for repair.
5	She has fallen 12 times in the past year. She was diagnosed with osteoporosis in 2006. Her [XXXXX] [XXXXX] density scan was in 2011. She has a history of RA. The patient has a low trauma / fragility fracture.
5	Fragility Fracture & [XXXXX] Health Consultation This consult is being performed at the request of Dr. [XXXXX] [XXXXX] to evaluate [PATIENT] [PATIENT] for fragility fracture and [XXXXX] health concerns. HPI [PATIENT] [PATIENT] is a 88 Y female who sustained a low energy fracture of the left IT femur on [5_19_19] after losing her balance and falling. Assessment The following risk factors exist for low [XXXXX] density : fragility fracture , Age $>$ 60 , female sex and sedentary lifestyle She takes the following medications / therapies which interfere with [XXXXX] quality : warfarin and SSR Is. Activity level is progressing , and she is working with therapy. Utilizes a [XXXXX] for ambulation.

Table 4.13 shows the results of predictions of classifiers trained on the training data for 5 and 7 sentence chunk notes alone and with all main features (note type, text length, Empath, MeSH anatomy terms) used.

Table 4.13: Test Data Predictions on 5 Sent. Notes and Notes with Note Type, Text Length, Empath and Anatomy Terms

Classifier	Features	Precision	Recall	F_1
GradientBoostingClassifier	Only Notes	**0.136**	**0.073**	**0.095**
DecisionTreeClassifier	Only Notes	0.107	0.065	0.080
GradientBoostingClassifier	Notes, Other Features	0.128	0.040	0.061
DecisionTreeClassifier	Notes, Other Features	0.088	0.040	0.055
Perceptron	Notes, Other Features	0.036	0.024	0.029
Perceptron	Only Notes	0.033	0.024	0.028
SVC	Only Notes	0.056	0.016	0.025
SVC	Notes, Other Features	0.056	0.016	0.025
MultinomialNB	Notes, Other Features	0.024	0.016	0.019
MultinomialNB	Only Notes	0.017	0.016	0.017
MLPClassifier	Notes, Other Features	0.017	0.008	0.011
RandomForestClassifier	Only Notes	nan	0.000	nan
KNeighborsClassifier	Only Notes	0.000	0.000	nan
LogisticRegression	Only Notes	0.000	0.000	nan
MLPClassifier	Only Notes	0.000	0.000	nan
RandomForestClassifier	Notes, Other Features	nan	0.000	nan
KNeighborsClassifier	Notes, Other Features	0.000	0.000	nan
LogisticRegression	Notes, Other Features	0.000	0.000	nan

Table 4.14 shows the results of predictions of classifiers trained on the training data for 5 and 7 sentence chunk notes alone and with all main features (note type, text length, Empath, MeSH anatomy terms) used, with the positive samples oversampled four times.

Table 4.14: Test Data Predictions on 5 Sent. Notes and Notes with Note Type, Text Length, Empath and Anatomy Terms With Oversampling of 4

Classifier	Features	Precision	Recall	F_1
DecisionTreeClassifier	Notes, Other Features	0.135	**0.121**	**0.128**
DecisionTreeClassifier	Only Notes	0.125	**0.121**	0.123
GradientBoostingClassifier	Only Notes	0.178	0.065	0.095
GradientBoostingClassifier	Notes, Other Features	**0.190**	0.032	0.055
MultinomialNB	Only Notes	0.026	0.056	0.036
MultinomialNB	Notes, Other Features	0.027	0.056	0.036
SVC	Only Notes	0.056	0.016	0.025
LogisticRegression	Only Notes	0.051	0.016	0.025
SVC	Notes, Other Features	0.056	0.016	0.025
Perceptron	Notes, Other Features	0.035	0.016	0.022
Perceptron	Only Notes	0.013	0.008	0.010
RandomForestClassifier	Only Notes	nan	0.000	nan
KNeighborsClassifier	Only Notes	0.000	0.000	nan
MLPClassifier	Only Notes	0.000	0.000	nan
RandomForestClassifier	Notes, Other Features	nan	0.000	nan
KNeighborsClassifier	Notes, Other Features	0.000	0.000	nan
LogisticRegression	Notes, Other Features	0.000	0.000	nan
MLPClassifier	Notes, Other Features	0.000	0.000	nan

Figure 4.7 is a visualization of the root of a decision tree classifier trained on 5 sentence chunks using only notes as a feature with an oversampling of 4.

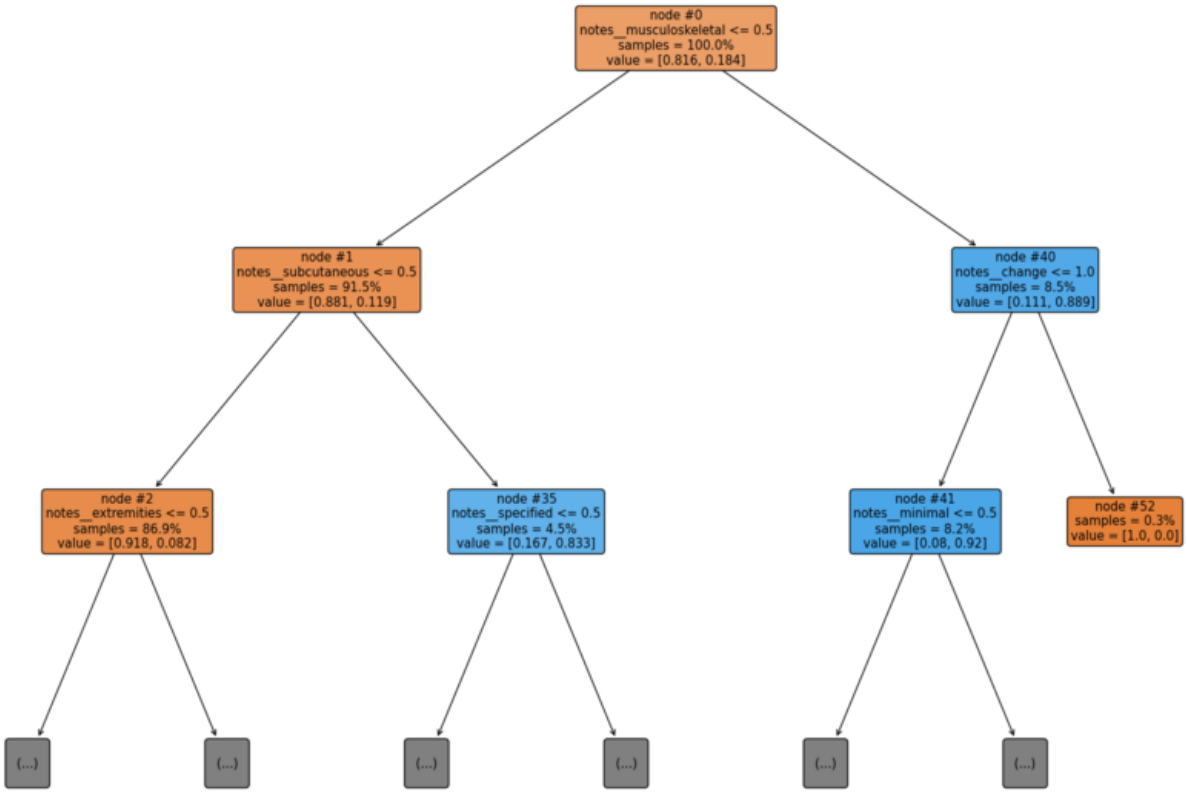

Figure 4.7: Section of Decision Tree for 5 Sent. Notes with Oversampling of 4

4.4 Discussion

Cross-validation (10-fold in all cases) results were collected across many iterations of different classifiers, features, text processing, and sentence chunk sizes. I chose to first find the best performing sentence chunk sizes across all classifiers and series of features. As shown in Figure 4.5, chunks of five sentences were the most common in the top 50% of results by F_1 score. 7 and 9 sentence chunks also performed decently, while 1 and 3 sized chunks did not. These results appear in tune with readings of the clinical notes, which showed relevant information about sarcopenia generally occurred a few sentences or so around the word.

Different iterations of features and text transformations were then cross-validated on all classifiers using chunks of five and seven sentences only, to narrow down the results on best performing chunk sizes found previously. There were no outliers in 1, 3 or 9 sentence chunk sizes that appeared significant to investigate further.

The clinical notes themselves alone are the first looked at detail as a baseline, with results shown in Table 4.3. Text transformations were then looked at, with notes as bigrams tested in Table 4.4 and NER transformation with the scispaCy "en_core_sci_sm" model in Table 4.5. Neither had a huge impact on F_1, but some classifiers scores were slightly improved. Additional text transformations not listed in the results were looked at, but performed worse than what is shown. A combination of unigrams and bigrams was tested but performed overall worse than bigrams alone. Ignoring default stopwords (the 'english' set as provided by sklearn) or ignoring terms with a very high frequency (generally, stop words automatically found based on the text) seemed to give the SVC classifier a slight improvement but overall little change (or significantly worse) for the others. Three other scispaCy models were tried ("en_core_sci_lg", "en_core_sci_scibert", and "en_ner_bionlp13cg_md") but all performed similar or worse.

Each feature then described in the Methodology was analyzed one by one, then all together, with results shown in Tables 4.6 - 4.10. None of the four features tested had any significant impact on the performance measures.

A simple oversampling was also tested in an attempt to shore up deficiencies of having a small dataset. The cross-validated results of an oversampling size of 4 (where positively coded samples are duplicated four times in the training data) are shown in Table 4.11. An oversampling size of 2 performed similarly. These results dramatically improve the performance measures, but a large part of this is likely due to overfitting.

Overall, in cross-validation, SVC tends to perform the best across F_1, with a relatively high precision and roughly 40-50% recall. Multinomial naïve Bayes performs the best across recall in all cases (save oversampling), typically between 0.5 and 0.6, but performing best with bigrams, with a recall of 0.897.

4.4.1 Predictions on Test Dataset

The test data manual rating resulted in 89 chunks being rated above 0 by at least one rater, and 614 chunks rated 0 by all raters. The above 0 averages were mostly grouped around 1 and 2, as 3 ratings occurred much more than 5 and even then did not always agree with the other raters. Only one rater classified any chunks with 5 (as well as 0 and 3). The other two only rated with 0 and 3. The distributions for these ratings, averaged over all three raters, is shown in Figure 4.6.

Predictions were performed on the test dataset on models trained on only the clinical notes, trained on the notes and the four main features, and trained on those combinations with an oversampling of 4. Results are shown in Table 4.13 and Tables 4.14. Performance measures shown were calculated by combining the results of predictions on all three ratings. Weighting of ratings of 5 was tried but made little impact (as only one rater used 5's). Using averages of all three sets was done as well, both with anything above 0 being positive, and anything above 1 being positive. This average showed very similar results to the approach of combining all three sets, as the variance between the percentages of true positives, etc are very small. These combinations all performed very poorly, with the best performance seen being decision tree classification with oversampling with an F_1 score of 0.128.

Bigrams, NER transformation, stopwords, and tf-idf were all tested with and without oversampling on the test dataset as well and all had worse or very similar performance as the results shown.

The performance on the test dataset is likely due to a few factors. The small sizes of the datasets can lead to not enough common language between the two to help identify discussion related to sarcopenia. The test dataset notes had very few examples that seemed to overwhelmingly suggest sarcopenia, based on the ratings received. Having no single unified test set also can make consolidating disagreeing information from multiple sets a challenge. These factors do show how different approaches may be taken to possibly find better results.

4.4.2 Decision Tree Analysis

An excerpt of a decision tree on a sentence chunk size of 5 on notes alone with oversampling of 4 is shown in Figure 4.7. While not the best in terms of cross-validation, decision trees performed decently compared to other classifiers overall and best (although still very poorly) on test predictions. The except here is shown to illustrate that it still appears to be on the right track, in that the most important word it selects on is 'musculoskeletal' which would be a very important word in regards to sarcopenia. Further analysis of various decision trees shows some words that might possibly relate, but lots of what appear to be noise as well, likely due to the limited data set size.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This research aimed to investigate prediction of the term 'sarcopenia' in clinical note text via machine learning. Various machine learning algorithms combined with different features and text processing was used to evaluate prediction of occurrence of the term in a small dataset of clinical notes provided by the Medical College of Wisconsin. The research showed, of the configurations of these tested, that none performed exceptionally well, based on F_1 score, but in many cases a support vector machine based model did show promise (better than 0.6 F_1 in cross-validation results). Other algorithms were of some interest as well in certain cases, notably decision tree, gradient boosting and logistic regression classifiers. Overall, the features used seemed to make very little difference. Oversampling had a very significant impact, but this is likely due to overfitting.

I believe we can safely draw the conclusion that the size and composition of the datasets is a large hindrance in producing a successful model for a task such as this. Compared to some of the related studies mentioned, which also found good performance with support vector machines and decision trees in certain cases, ideas in this study may roughly be on a promising track, but just lack the quantity and type of data. Data that is more concerned

with sarcopenia's details, in the sense of a diagnosis or screening test, may provide much more helpful text to assist in building a model. This could help offset issues in this study where a mere mention of sarcopenia in a note with little or no context can sway the model too much due to the small dataset size. The features used may show more promise as well with better data, especially the Empath and MeSH related ones, if they have more content to match on.

5.2 Future Work

The process of developing these experiments showed a plethora of future directions work on this topic could go in. In addition to many novel libraries being released for spaCy, there is also another library oriented around NLP and text processing: Spark NLP. More interestingly for this specific topic exists Spark NLP for Healthcare, a NLP library oriented around analyzing clinical data. An academic license was requested for experimentation but was not obtained in time to be able to make use of it. A preliminary analysis suggests the models available via this library are worth examining for problems as this.

There are also many opportunities to process and augment the existing data in new ways. Alternate forms of chunking such as by character or word count or building an alternate sentence tokenizer based around the clinical notes themselves could provide a better window into relevant text regions. Topic modeling could be useful as well, especially if trained on a larger dataset or relevant data pertaining to sarcopenia. Alternative sampling techniques for small datasets and more automated parameter tuning across different classifiers is also worth exploring. Finding a way to utilize phenotypes built up from clinical records for sarcopenia as mentioned in Related Work is another good idea for future work.

Of course, as stated above, a greater amount and variety of clinical notes pertaining to sarcopenia would be a likely be the most promising way to continue analyzing this problem. More data to build models of that includes more description of patients with sarcopenia from different clinical viewpoints would likely help build much more robust models.

Bibliography

[1] Alfonso J Cruz-Jentoft and Avan A Sayer. Sarcopenia. *The Lancet*, 393(10191):2636–2646, 2019.

[2] World Health Organization, editor. *International statistical classification of diseases and related health problems*. World Health Organization, Geneva, 10th revision, 2nd edition edition, 2004.

[3] Li Cao and John E. Morley. Sarcopenia is recognized as an independent condition by an international classification of disease, tenth revision, clinical modification (icd-10-cm) code. *Journal of the American Medical Directors Association*, 17(8):675–677, 2016.

[4] Bureau of Labor Statistics, U.S. Department of Labor. Occupational outlook handbook, medical records and health information specialists. `https://www.bls.gov/ooh/healthcare/medical-records-and-health-information-technicians.htm`. [Online; accessed: 2021-08-14].

[5] U.S. Department of Health & Human Services. The Health Insurance Portability and Accountability Act of 1996 (HIPAA). `https://www.cdc.gov/phlp/publications/topic/hipaa.html`. [Online; accessed: 2021-08-14].

[6] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 3 edition, 2009.

[7] J. Eisenstein. *Introduction to Natural Language Processing*. Adaptive Computation and Machine Learning series. MIT Press, 2019.

[8] Ethan Fast, Binbin Chen, and Michael S. Bernstein. Empath: Understanding topic signals in large-scale text. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, page 4647–4657, 5 2016.

[9] Itika Gupta, Barbara Di Eugenio, Devika Salunke, Andrew Boyd, Paula Allen-Meares, Carolyn Dickens, and Olga Garcia. Heart failure education of african american and hispanic/latino patients: Data collection and analysis. In *Proceedings of the First Workshop on Natural Language Processing for Medical Conversations*, pages 41–46, 7 2020.

[10] Yla R. Tausczik and James W. Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of Language and Social Psychology*, 29(1):24–54, 2009.

[11] CE Lipscomb. Medical subject headings (MeSH). *Bulletin of the Medical Library Association*, 88(3):265—266, July 2000.

[12] U.S. National Library of Medicine. 2019 MeSH files. `https://wayback.archive-it.org/org-350/20191102205610/https://www.nlm.nih.gov/mesh/2019/download/2019New_Mesh_Tree_Hierarchy.txt`. [Online; accessed: 2021-08-10].

[13] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.

[14] Seyedmostafa Sheikhalishahi, Riccardo Miotto, Joel T Dudley, Alberto Lavelli, Fabio Rinaldi, and Venet Osmani. Natural language processing of clinical notes on chronic diseases: Systematic review. *JMIR Medical Informatics*, 7(2):e12239, 2019.

[15] Travers Ching, Daniel S. Himmelstein, Brett K. Beaulieu-Jones, Alexandr A. Kalinin, Brian T. Do, Gregory P. Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M. Hoffman, Wei Xie, Gail L. Rosen, Benjamin J. Lengerich, Johnny Israeli, Jack Lanchantin, Stephen Woloszynek, Anne E. Carpenter, Avanti Shrikumar, Jinbo Xu, Evan M. Cofer, Christopher A. Lavender, Srinivas C. Turaga, Amr M. Alexandari, Zhiyong Lu, David J. Harris, Dave DeCaprio, Yanjun Qi, Anshul Kundaje, Yifan Peng, Laura K. Wiley, Marwin H. S. Segler, Simina M. Boca, S. Joshua Swamidass, Austin Huang, Anthony Gitter, and Casey S. Greene. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387, April 2018.

[16] Wei-Qi Wei, Pedro L Teixeira, Huan Mo, Robert M Cronin, Jeremy L Warner, and Joshua C Denny. Combining billing codes, clinical notes, and medications from electronic health records provides superior phenotyping performance. *Journal of the American Medical Informatics Association*, 23(e1):e20–e27, April 2016.

[17] Yanshan Wang, Liwei Wang, Majid Rastegar-Mojarad, Sungrim Moon, Feichen Shen, Naveed Afzal, Sijia Liu, Yuqun Zeng, Saeed Mehrabi, Sunghwan Sohn, and Hongfang Liu. Clinical information extraction applications: a literature review. *Journal of Biomedical Informatics*, 77:34–49, 2018.

[18] Kasper Jensen, Cristina Soguero-Ruiz, Karl Oyvind Mikalsen, Rolv-Ole Lindsetmo, Irene Kouskoumvekaki, Mark Girolami, Stein Olav Skrovseth, and Knut Magne Augestad. Analysis of free text in electronic health records for identification of cancer patient trajectories. *Scientific Reports*, 7(1):46226, May 2017.

[19] Guergana K Savova, Jin Fan, Zi Ye, Sean P Murphy, Jiaping Zheng, Christopher G Chute, and Iftikhar J Kullo. Discovering Peripheral Arterial Disease Cases from Radiology Notes Using Natural Language Processing. page 5.

[20] Carol Friedman, Thomas C. Rindflesch, and Milton Corn. Natural language processing: State of the art and prospects for significant progress, a workshop sponsored by the National Library of Medicine. *Journal of Biomedical Informatics*, 46(5):765–773, October 2013.

[21] Wei-Hung Weng, Kavishwar B. Wagholikar, Alexa T. McCray, Peter Szolovits, and Henry C. Chueh. Medical subdomain classification of clinical notes using a machine learning-based natural language processing approach. *BMC Medical Informatics and Decision Making*, 17(1):155, 2017.

[22] Guhan Ram Venkataraman, Arturo Lopez Pineda, Oliver J. Bear Don't Walk IV, Ashley M. Zehnder, Sandeep Ayyar, Rodney L. Page, Carlos D. Bustamante, and Manuel A. Rivas. FasTag: Automatic text classification of unstructured medical narratives. *PLOS ONE*, 15(6):e0234647, 2020.

[23] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, 2014. Association for Computational Linguistics.

[24] Dina Demner-Fushman, Willie J Rogers, and Alan R Aronson. MetaMap Lite: an evaluation of a new Java implementation of MetaMap. *Journal of the American Medical Informatics Association*, 24(4):841–844, 2017.

[25] Yanshan Wang, Sunghwan Sohn, Sijia Liu, Feichen Shen, Liwei Wang, Elizabeth J. Atkinson, Shreyasee Amin, and Hongfang Liu. A clinical text classification paradigm using weak supervision and deep representation. *BMC Medical Informatics and Decision Making*, 19(1):1, 2019.

[26] Naveed Afzal, Sunghwan Sohn, Sara Abram, Christopher G. Scott, Rajeev Chaudhry, Hongfang Liu, Iftikhar J. Kullo, and Adelaide M. Arruda-Olson. Mining peripheral arterial disease cases from narrative clinical notes using natural language processing. *Journal of Vascular Surgery*, 65(6):1753–1761, 2017.

[27] Hongfang Liu, Suzette J. Bielinski, Sunghwan Sohn, Sean Murphy, Kavishwar B. Wagholikar, Siddhartha R. Jonnalagadda, K. E. Ravikumar, Stephen T Wu, Iftikhar J. Kullo, and Christopher G. Chute. An information extraction framework for cohort identification using electronic health records. Mar 2013.

[28] Ranjani N. Moorthi, Ziyue Liu, Sarah A. El-Azab, Lauren R. Lembcke, Matthew R. Miller, Andrea A. Broyles, and Erik A. Imel. Sarcopenia, frailty and cachexia patients detected in a multisystem electronic health record database. *BMC Musculoskeletal Disorders*, 21(1):508, 2020.

[29] Paul Braunschweiger and Kenneth W. Goodman. The CITI Program: An International Online Resource for Education in Human Subjects Protection and the Responsible Conduct of Research:. *Academic Medicine*, 82(9):861–864, September 2007.