A Privacy-Preserving Natural Language Clinical Information Extraction pipeline

by

Jeffrey Agnel Pinto

A Research Paper submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Abstract

A Privacy-Preserving Natural Language Clinical Information Extraction pipeline

Jeffrey Agnel Pinto
Master of Science
Graduate Department of Computer Science
University of Toronto
2019

Automated clinical information extraction (IE) from scanned, unstructured clinical psychiatric notes can be improved beyond the state-of-the-art, rule-based tools by using Natural Language Processing (NLP) Machine Learning (ML) models, transfer learning and features derived from word embeddings.

For smaller research teams, current tools can be too complicated to implement, too difficult to query, or unable to handle the variety of source documents, resulting in clinical IE being a manual process. In contrast, NLP ML models, which rely on large volumes of data to perform well, are challenged by the small sample sizes of many projects. We present a pipeline of customizable modules that allows users to locally query raw source documents, ensuring 'privacy by design'. This approach allows modules to be easily replaced, upgraded, and augmented with supplemental data.

We show that, through the use of unsupervised learning of word embeddings and semi-supervised expansion of training data, this approach out-performs similar expert systems with the same amount of configuration. Our approach is particularly useful in reducing the volume of notes that researchers manually review, where identifying positive samples of query terms is the goal. In some cases, this pipeline achieved a 92% rate of predicting positive samples.

Finally, this approach encourages a cycle of continual improvement by allowing users to securely transfer private data between restricted projects, share de-identified data between teams, and incrementally improve performance of ML algorithms as more relevant data is made available.

Acknowledgements

1. Dr. Graeme Hirst, Professor, Department of Computer Science, University of Toronto

2. Dr. James Kennedy, Head, Molecular Science and Head, Tanenbaum Centre for Pharmacogenetics, Campbell Family Mental Health Research Institute, CAMH

3. Dr. John Strauss, Clinician Scientist, Child, Youth and Family Program, CAMH

4. Sheraz Cheema, Research Analyst, CAMH

5. Anashe Shahmiriam, Clinical Research Analyst, CAMH

6. Nicole Braganza, IMPACT Project Manager, CAMH

7. Serena Jeblee, Department of Computer Science, University of Toronto

8. With gratitude: Mom, Dad, Giles, Marlene, Gabby, Ellis, Mike, Stan, Mahen, Janet, Lily, Amber, and Aaron.

Contents

1 Introduction **1**

1.1 Motivation to Automate 1

1.2 Constraints on Machine Learning 2

1.3 Research Objective 3

2 System Design **4**

2.1 Baseline Clinical Information Extraction 4

2.2 The IMPACT study 5

2.3 A Modular NLP Pipeline 6

3 Process Noisy Source Data **9**

3.1 From Scan to Text 9

3.2 Text Pre-Processing 12

3.3 Label Pre-Processing 14

4 Feature Extraction **16**

4.1 Word Embeddings 16

4.2 POS Tags 17

5 Data Expansion **18**

5.1 Training Data Balance 18

5.2 Synthentic Resampling 20

5.3 Record to Sentence Labelling 20

6 Algorithm Selection **23**

6.1 Training Segmentation 23

6.2 Evaluation 25

7 Results **27**
7.1 Comparison of Results 27
7.2 Limitations 31

8 Discussion and Further Work **32**

Bibliography **34**

Chapter 1

Introduction

1.1 Motivation to Automate

A common and expensive bottleneck in medical research is reviewing confidential patient medical records. This is necessary to select research candidates [Geraci *et al.*, 2017], extract relevant patient medical data, and document clinical results [Herbert *et al.*, 2018]. The recent, wide-spread adoption of Electronic Health Records (EHRs) in general care clinics and the ongoing standardization of privacy protocols to transfer information has meant rapid growth in the availability of patient data to many clinical research institutions [Legislative Assembly of the Province of Ontario, 2000; US Department of Health and Human Services, 2018]. However, the tools to evaluate the unstructured portions of the EHRs at scale are often expensive to configure, specialized to unrelated tasks, or too generalized to be accurate to specific research goals [Wang *et al.*, 2017]. The result is that most researchers perform only simple keyword searches or use only structured data like diagnosis fields when evaluating candidates.

Structured clinical information extraction is often insufficient because critical insights lie in the unstructured text, as this is where a clinician communicates the majority of their qualitative evaluations. Consider the recruitment of research candidates for clinical studies. In the case of pharmaceuticals, an extra month of delay can cost up to US$25M in potential income [Marks & Power, 2002]. Traditional methods of candidate selection are a bottleneck as they can fail to identify up to 60% of possible participants [Fink *et al.*, 2004]. These methods are mostly a manual process of reviewing patient medical records, balancing candidate profile groups, and conducting screening interviews.

As well, structured data analysis alone can be problematic as this data may be missing, too ambiguous, or even erroneous and contradictory to the unstructured text. Researchers on a psychiatric study used a large training corpus to show that an NLP model

can be more than 50% more accurate than structured data analysis alone [Perlis *et al.*, 2012]. They improved the quality of analysis by applying general machine learning (ML) algorithms to specific specialist domains. Similarly, mental health researchers in the UK have begun using EHRs to aid in research recruitment by increasing the number of potential participants and reducing the burden of manual screening by clinicians [Callard *et al.*, 2014; Ross *et al.*, 1999]. These examples share the common trait of using ML to evaluate natural language inputs in EHRs and output medically relevant content that can be used to classify psychiatric candidates [Gonzalez-Hernandez *et al.*, 2017].

1.2 Constraints on Machine Learning

State-of-the-art supervised ML models are typically trained on large data sets with the assumption that training and test data are independently and identically distributed. These models tend to overfit to the majority class when trained on limited data, exhibiting a *sample selection bias* that was introduced implicitly during training. This is a significant problem in domains like clinical health. Training sets are expensive to annotate, and so are often small — in the order of hundreds or thousands as opposed to millions or billions of records.

Further, clinical notes are often extracted from a variety of EHRs, leading to a heterogeneity of structure with records including headings, field names or other text artifacts from the EHRs. These notes might also be stored only in printed format (e.g. pdfs), which results in additional text artifacts / noise being introduced during the optical character recognition (OCR) process. Thus, machine learning approaches must also address the cumulative noise and, "that two kinds of errors are lumped together" [Olieman *et al.*, 2017] when extracting information in a practical setting.

These kinds of errors are an issue not only for overall performance but also in meeting protected health information privacy laws. Standard practice with clinical data is to de-identify records by replacing certain types of tokens (e.g., proper nouns, dates, locations) with anonymous placeholder text [Neamatullah *et al.*, 2008]. If notes contain too much noise, the initial part-of-speech tagging fails to correctly identify all private information. Health research guidelines do not allow any private information to be released, so, given a long enough set of notes, this guideline cannot be confidently met. However, perhaps due to relatively recent increases in interest, there is only limited research on privacy-preserving computing (PPC) mechanisms of NLP artifacts [Malik *et al.*, 2012; Gardner & Xiong, 2009; Cavoukian, 2011]

1.3 Research Objective

The goal of this work is to implement a "good niche application that really [does] have value" per the desiderata laid out by Church & Hovy [1993, pg. 246]:

(a) it should set reasonable expectations,

(b) it should make sense economically,

(c) it should be attractive to the intended users,

(d) it should exploit the strengths of the machine and not compete with the strengths of the human,

(e) it should be clear to the users what the system can and cannot do, and

(f) it should encourage the field to move forward toward a sensible long-term goal.

To meet these criteria, we present a modular, end-to-end clinical information extraction pipeline to evaluate the effectiveness at multiple stages of current best practice NLP approaches versus the more commonly used rule-based systems. We provide an overall design and examples of an evolutionary framework that integrates existing private, unstructured notes, expert systems, and NLP algorithms to allow users to quickly query clinical notes using personalized search terms without having to perform complex configuration tasks. This framework can be deployed as a local application, thus allowing access control on a user-by-user basis to view the protected health information contained in source and supplemental data.

Specifically, this pipeline is designed to resolve the following five issues:

1. Complexity: users perform minimal configuration tasks in order to query raw text using any keywords or concepts.

2. Unstructured text: the pipeline evaluates state-of-the-art statistical NLP ML algorithms to extract key elements from raw text.

3. Small sample size: the pipeline uses transfer learning to allow users to easily incorporate additional data sources.

4. Noisy input: the pipeline's modular design allows users to customize the types of corrections and de-identification to apply to their data.

5. Privacy: the pipeline's architecture supports local deployment, which allows data to be annotated only by trusted users.

Chapter 2

System Design

2.1 Baseline Clinical Information Extraction

A comprehensive overview of clinical information extraction identified a number of tools from current research [Wang *et al.*, 2017]. These systems, although built on extensible frameworks, rely heavily on rule-based approaches as opposed to using statistical machine learning approaches which are currently dominant in NLP research. In addition, many tools are specialized to extract only certain concepts (e.g. MedEx and MedXN are focused medication IE).

Part of the reason for this incongruity may be the easier interpretability of a rule-based system, as it is trivial to identify the specific phrases that resulted in an extracted data point [Vellido *et al.*, 2012]. Additionally, rule-based systems allow users explicit insertion of domain-specific knowledge and adaptation of existing rules as needed.

For this project, we used the clinical Text Analysis and Knowledge Extraction System (cTAKES) default clinical pipeline [Savova *et al.*, 2010] in order to contrast our results with a representative, rule-based IE system. Wang *et al.* [2017] identified this as the most cited IE tool across 263 studies published since 2009. When applied to clinical notes, it creates linguistic and semantic annotations by executing components in sequence to process the clinical notes. Components may rely on other components' output in order to produce the final annotation, which may make this tool susceptible to noisy input data as errors in initial components get magnified by subsequent ones.

In particular, we use the output of the cTAKES Named Entity Recognition (NER) component as the predicted labels for our clinical notes. cTAKES NER uses a dictionary that is a subset of the Unified Medical Language System (UMLS) [Ogren *et al.*, 2007]. Each term in the dictionary belongs to one of the following semantic types: Anatomical sites, Signs/Symptoms, Procedures, Anatomy, Diseases/Disorders, and Medications [Bo-

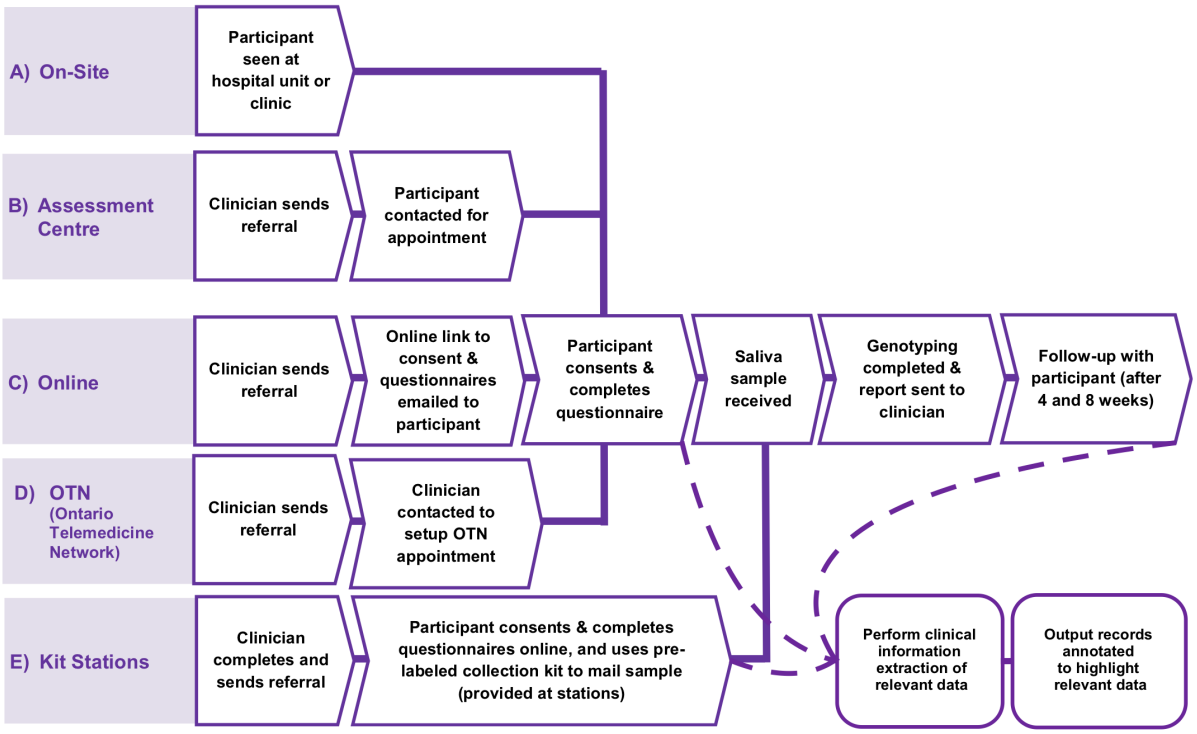


Figure 2.1: Flowchart illustrating the process involved in recruiting participants into the IMPACT study and showing the steps where clinical IE is needed. Adapted from Herbert *et al.* [2018]

denreider & McCray, 2003]. This project uses the Signs/Symptoms, Diseases/Disorders, and Medications types as baseline predictions for test data.

2.2 The IMPACT study

For this project, we utilize a practical data set from a pharmacogenetic study at the Centre for Addiction and Mental Health in Toronto. The Individualized Medicine: Pharmacogenetics Assessment and Clinical Treatment (IMPACT) project is a seven-year, ongoing province-wide research study that aims to optimize pharmacological treatment for mental health patients. The goal of IMPACT is to increase the success rate of drug response and medication adherence, and to reduce the risk of side effects from medications. IMPACT provides guidance for choosing medication based on individual patient genotype and clinical symptoms, which are obtained via a saliva sample, interviews, questionnaires and EHRs. As of September of 2018, more than 11,000 participants have enrolled in the study.

The analysis is conducted as shown in *Figure 2.1*. Data extracted from participants'

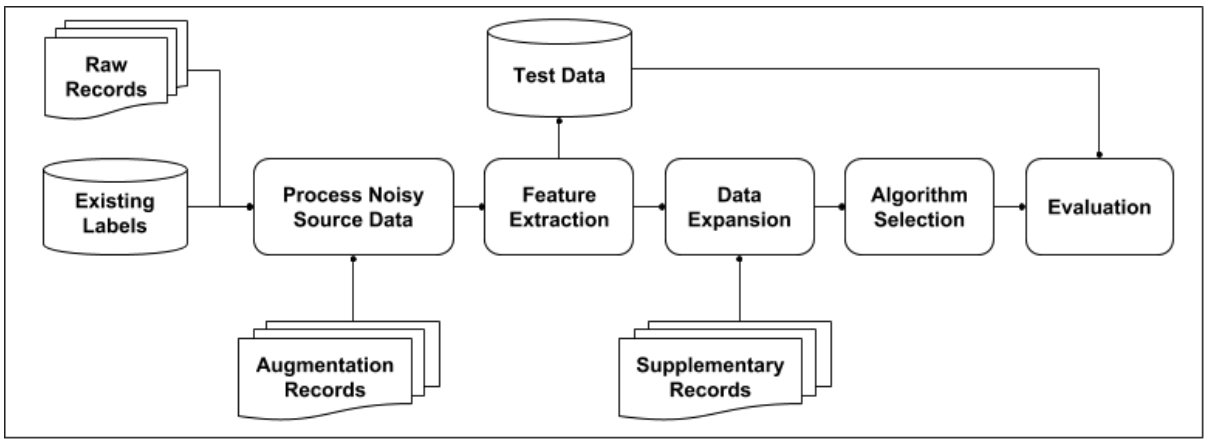


Figure 2.2: Diagram of the IMPACT NLP IE Pipeline; for this project, the processing module was augmented with PubMed unigrams for spell-checking [Maloney *et al.*, 2013] and the data expansion module was supplemented with MIMIC-III clinical notes [Johnson *et al.*, 2016]. Details of each step can be found in following chapters.

medical records is merged with baseline and follow-up data from questionnaires. Clinical IE is used to extract relevant medication, condition/symptom and adverse events/side effects from each source. At study completion, a more comprehensive analysis of all data collected will be reported. A key issue for IE is the naturalistic design of IMPACT; most questionnaire and EHR data is entered into free-entry text fields so there is little standardization in structure or content from record to record and, indeed, within a single participant's record collection [Herbert *et al.*, 2018].

2.3 A Modular NLP Pipeline

We present an NLP pipeline as shown in *Figure 2.2*. All the modules are composed of open source or freely available NLP software packages. In addition, by chaining modules together, the pipeline can be easily extended, tuned and re-evaluated with minimal configuration changes by modifying or inserting individual modules.

To begin, we collect the **Raw** documents for participants that have had existing annotations made by IMPACT staff. Annotations are at the participant level and do not mention *cue words* or specific locations in documents. This provides the basis for our supervised learning approach with annotations converted into *multihot* labels for each participant.

Both medical records and labels are then **Processed** to aggregate data by participant into a single record, correct spelling mistakes, de-identify private health information, and identify label synonyms. For spelling correction, we use a custom medical lexicon based

on the labels and publicly available abstracts and content for research papers referenced in PubMed [Maloney *et al.*, 2013].

Note that although annotators group labels into 3 classes (adverse events, condition, and medication) this is primarily for subsequent analysis. Each participant can have i labels, $1 \leq i \leq n$, where n is the total count of all labels.

The resulting *gold standard corpus* (GSC) is then passed to our **Feature Extraction** step. Here, we convert text data to a vectorized format representing both the entire participant's text corpus as well as calculating a vector for each sentence in that corpus. In addition, we extract part-of-speech (POS) tags for each text vector and append the frequency counts based on the occurrence of the 33 Penn Treebank POS tags [Santorini, 1990].

As this is a multi-label task with limited samples, simply holding out a fixed set of records would not allow us to evaluate performance across all labels, as a fixed set would not contain positive examples of all labels. We extract a subset of the GSC as held-out **Test Data** with each label term having a unique data set composed of at least one positive sample for each label, even if this is the only positive sample in the GSC.

Training data and features are passed to the **Data Expansion** module. Here we supplement and resample our data set to improve classification results in several ways. We apply *transfer learning* by adding clinical notes similar to our data from the MIMIC-III freely accessible critical care database [Johnson *et al.*, 2016]. Similar notes are identified via n-gram matches of label terms or the similarity of training text in a shared word embedding space. Once we have supplemented the training data for each label we apply synthetic resampling techniques to further balance positive and negative classes.

Our **Algorithm Selection** module splits training data for each label into training and validation subsets for *cross-validation*. In total, 10 ML models that have achieved successes on this or related NLP tasks were evaluated: two support vector machines (SVMs) [Perlis, 2013], two perceptron networks, a convolutional neural network (CNN) [**?**], a long short-term memory recurrent neural network (LSTM RNN) [Hochreiter & Schmidhuber, 1997], a bidirectional LSTM RNN [Graves & Schmidhuber, 2005], a random forest, and a logistic regression classifier. As well, a baseline multinomial Naive Bayes classifier was evaluated. Additionally, when evaluating sentence text, as opposed to record text, we used a two-stage classification approach of regression then classification.

Finally during **Evaluation**, each label-specific classifier is applied to our test set at the record and sentence level. Sentence predictions are aggregated into a single prediction for a record. Thus, we are able to contrast the accuracy of IE when considering documents as-a-whole versus individual sentences. Additionally, test data is processed through cTAKES

to obtain a comparative benchmark with a traditional clinical IE approach.

In contrast to many ML applications, the goal of clinical classification of new records is asymmetrical. Users are trying to identify positive examples of the minority class — accurately identifying `bipolar disorder` in patients, which occurs in <1% of the population, is more important than accurately identifying those without the disorder [McDonald *et al.*, 2015]. Thus, a critical factor for evaluating model success will be the ability to identify probable positive label matches. This is because, due to practical time and budget constraints, only these records will be further manually reviewed. Records classified without label matches are not revisited. So, along with optimizing for the best performing model on *accuracy* and *F1* scores at the classification level, particular attention is paid to the *positive predictive* rate of the classifiers.

Chapter 3

Process Noisy Source Data

3.1 From Scan to Text

This project begins with **117** annotated records across **225** documents. Due to the naturalistic design of the IMPACT study, our data is heterogeneous with a mixture of scanned print outs, text files, and hand-written clinical notes as the samples in *Figure 3.1*. The volume of notes is also not evenly distributed among participants and, similarly, the volume does not dictate the number of labels associated with a record, as shown in *Figure 3.2*. We use as much input data as possible as the labels in our GSC were applied after manually reviewing all these data sources, and the majority of these labels occur within the GSC text, sometimes even in hand-written portions, as exact n-grams. So, rather than excluding noisy text, we clean it through a sequence of steps.

We initially convert all documents to sets of individual page images, then apply the open source Tesseract Optical Character Recognition (OCR) software on a page-by-page basis [Smith, 2007]. By applying OCR to each page separately, the software is able to adapt to changes in layout, font, and contrast. OCR works by converting image data to text character-by-character. The resulting text contains extra white space, non-alpha-numeric characters, misspelled words, and merged words depending on the quality of the source. OCR output is then appended to the record at the page level for subsequent cleaning.

The output for the **225** unique documents contains **34,695** sentences and also reaffirms the heterogeneity of the source documents. Records have an average length of **66,000** tokens but range from 2,884 to 382,102 tokens. Further details are shown *Table 3.1*

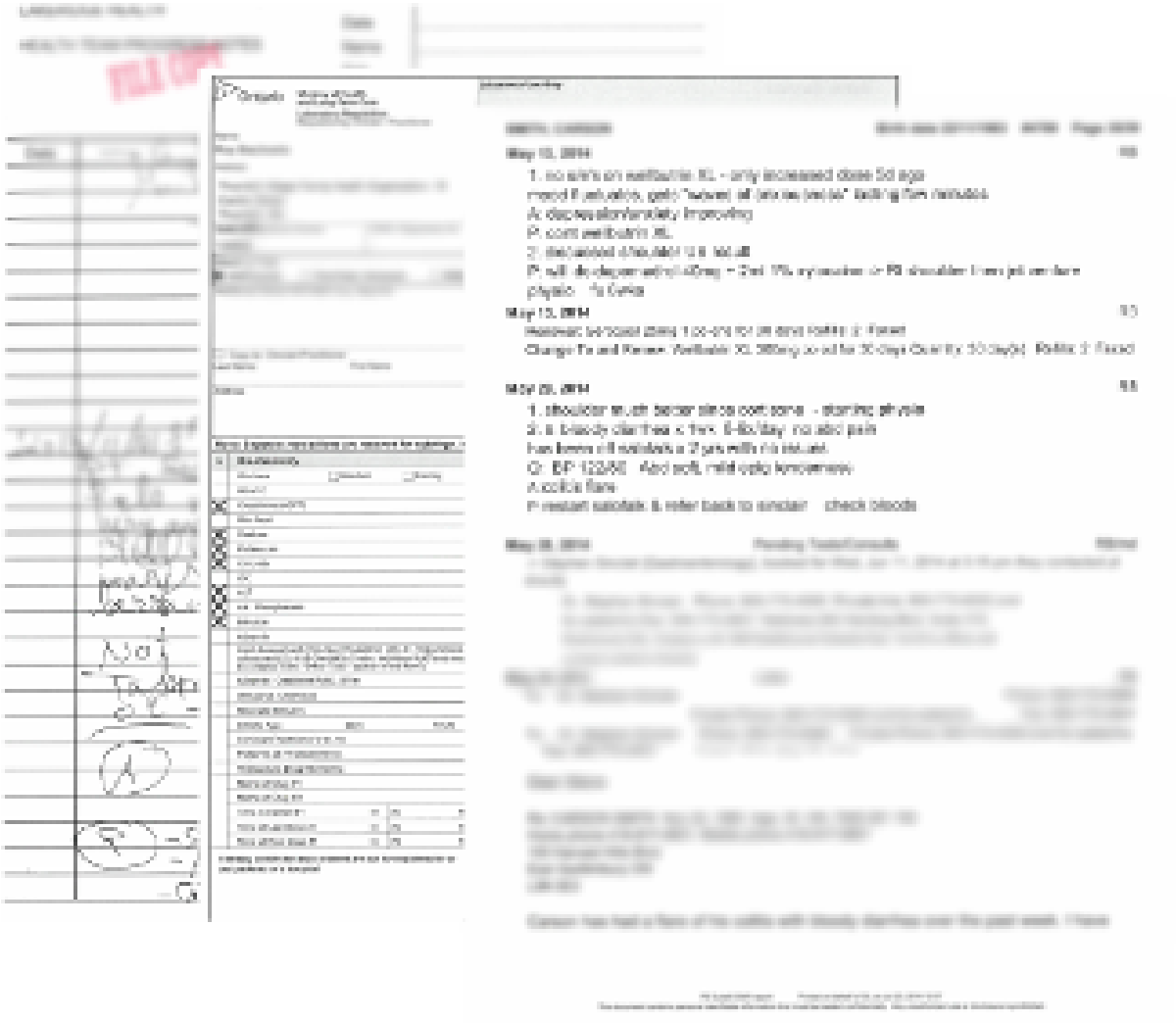


Figure 3.1: Sample raw source documents show a large variety of layout, formatting, and quality; the content is blurred for privacy

Table 3.1: Corpus Metrics for source data through pre-preprocessing

	Raw Data	**Post Spell-Check**	**Post DE-ID**	**Clean Text**
Token Count	8,317,992	8,317,992	8,157,658	7,545,180
Avg Length	66,313	66,313	65,133	62,062
Min Length	2,884	2,884	2,798	2,722
Max Length	382,102	382,102	378,011	371,424
Unique Tokens	102,074	87,928	85,560	81,682

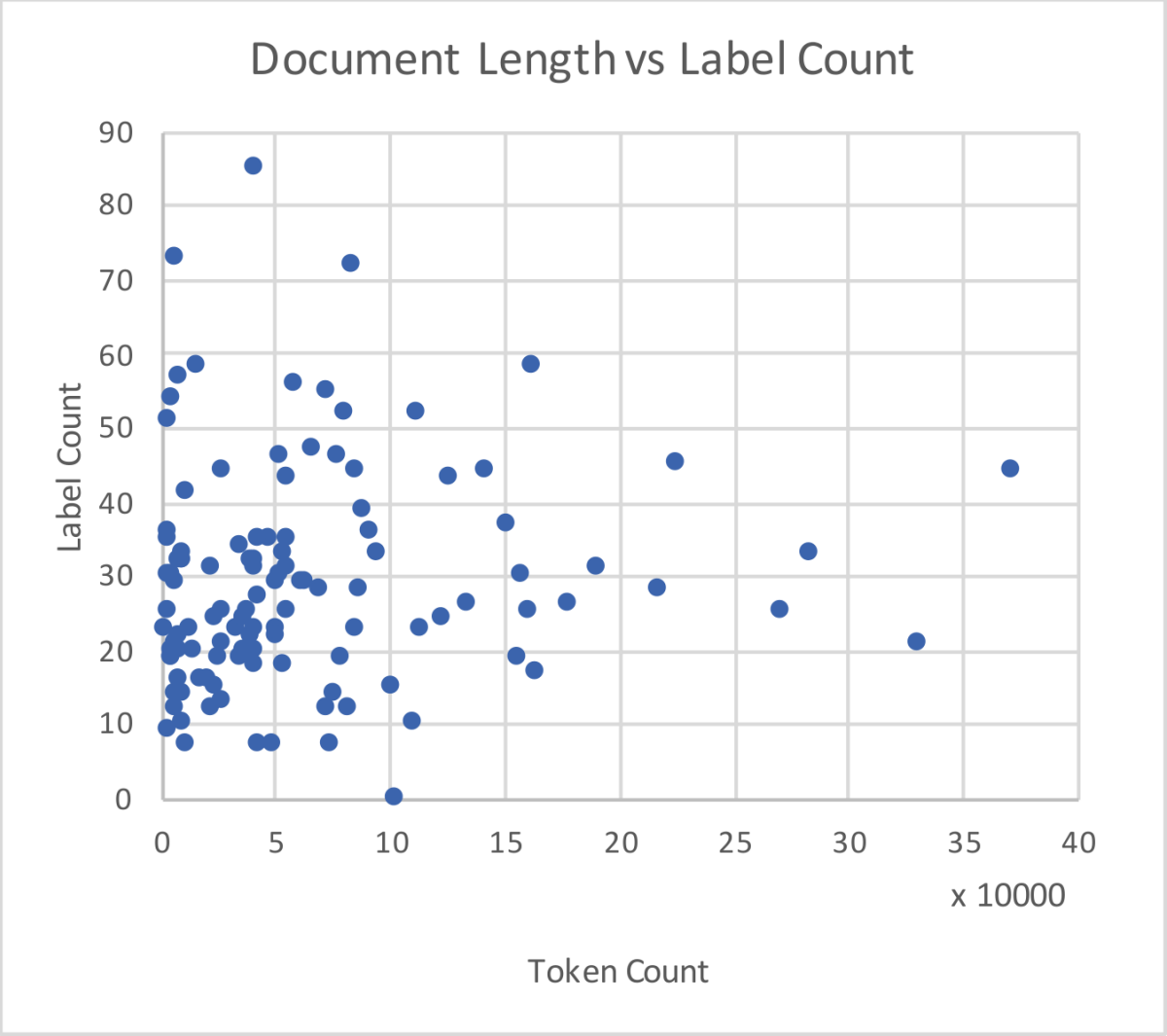


Figure 3.2: Plot of label counts vs record length shows no strong correlation of the two.

3.2 Text Pre-Processing

Once our source data is converted into plain text and aggregated by record, we can apply several layers of additional pre-processing. First, we build a reference medical lexicon, `MedLex`, with frequencies of unigrams extracted from PubMed abstracts and openly available articles. For efficiency, this project used the 24,000,000 unigrams made available by Moen & Ananiadou [2013]. We attempted to use the MIMIC-III data as an additional unigram source; previous research showed that these texts suffer from similar issues as our source data and thus introduce similar errors of merged and misspelled words [Fivez *et al.*, 2017].

With this lexicon we iterate all source tokens $[w_1, w_2, ..., w_m]$ where m is the number of tokens in the record. If w_1 is in our lexicon, no change is made and we process w_2 and so on. If w_i is not in the list, we return all valid token candidates C from the lexicon that have a maximum Levenshtein distance of 2 from our source token. Levenshtein distance is the number of deletions, insertions, or substitutions required to transform w into c. In the case of multiple lexicon tokens within our edit distance, we replace our source token with the token that had the highest frequency in PubMed data, or $w_i = argmax_{c \in C} P(c)$ [Norvig, 2007].

An example of spelling correction can be seen in the sentences below with only 2 of 3 possible changes made as the last element is too noisy to interpret:

```
HISTORY (0,9.  naurological.  respiratory, cagdiac, gnothcrissues)
HISTORY (0,9.  neurological.  respiratory, cardiac, gnothcrissues)
```

After spelling correction, in order to meet medical research requirements to protect PHI and legal restrictions imposed by legislation such as HIPAA, we process source text through `deid`, an open-source, Java de-identification process that has been approved by the hospital Research Ethics Board [Neamatullah *et al.*, 2008]. The de-identification process replaces most proper nouns such as names and locations as well as dates with anonymous keys linked to a separate index file that can be used to reconstruct data. To ensure our pipeline respects PHI, we store but do not use these index files.

Next, inspecting the training corpus reveals additional opportunities to reduce processing time and increase accuracy. The reduction in processing time is achieved by reducing the token count of each document as each model considers individual tokens. The accuracy assumption is considered below.

As visible in *Figure 3.3*, many of the most frequent tokens in the GSC were numeric (10 of top 50). This was a result of date-time entries in the text or the replacement

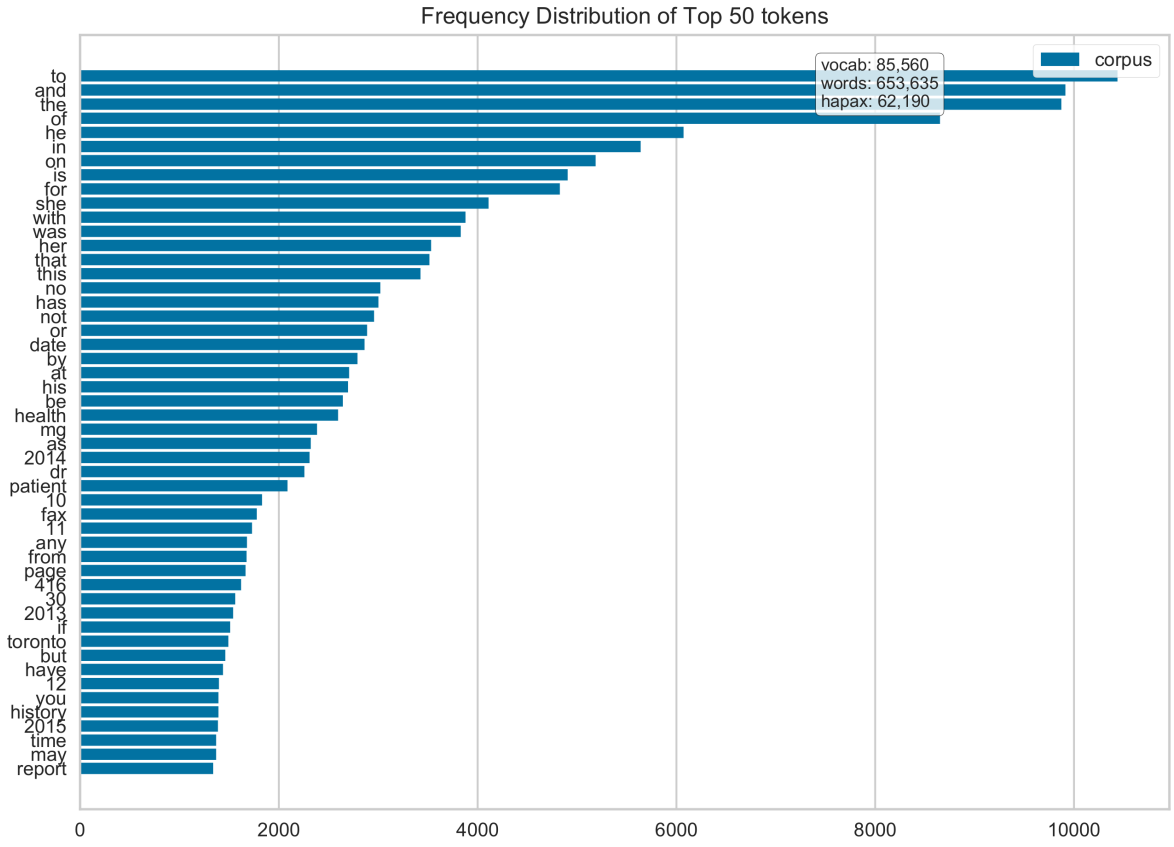


Figure 3.3: Token frequency before pre-processing.

tokens inserted by `deid`. In addition, many of these NE tokens are too specific as the de-identification process generated unique references within each document for named entities (NE) as opposed to applying a uniform token. This not only does little to reduce the unique token count it also introduces additional errors through inferring incorrect details such as confusing genders. An example of this can be seen in the document snippet below:

```
The arrangement now is that her parents alternate one week at a time living
in the home since about the end of [**2017-09-27**].  Her mother feels
that this is very disruptive for [**Male First Name (un) 1**].  We spoke
about individual therapy but [**Male First Name (un) 1**] initially stated
that she was not interested.
```

These de-identified tokens and numeric data have no influence on our event, condition, or medication labels. Thus, the initial step was to replace all data between `[ ]` with a simple token; either `digits` or `name` was used. Similarly, additional numeric data was replaced and all text was converted to lower case.

An additional step of using lemmatization or stemming [Porter, 1980] of words was evaluated early in the project, but rejected for several reasons. Both lemmatization and stemming attempt to convert the inflected form of words to a base form. The logic is that most inflected words are used in the same pragmatic way in text to convey meaning, especially in regards to sentiment. Thus, *a depressed person* means the same as *a person with depression*. However, for this project, the standard tools had trouble correctly transforming many medical terms that annotators used to classify the records; for example, the medical diagnosis of *depressive disorder* was transformed into *depress disorder* where *depress* is treated as a verb instead of an adjective.

The processing steps reduce the unique token (word) count in the GSC by 5% from 85,560 to 81,682 and the average document length by 4% from 65,133 tokens to 62,062 as shown in *Table 3.1*. This processed dataset is referred to as `Clean` data whereas the original data is referred to as `Raw` data.

3.3 Label Pre-Processing

We performed limited pre-processing to GSC labels, as the goal of the project is to classify documents when given only a naturalistic text input. We apply minor spelling correction manually to 36 of the 1500 labels such as relabelling `[tri cyclen-lo, tri-cyclen, tri-cyclen 28,` and `tri-cyclen lo]` to `tricyclen`. We do not, however, convert each label to a structured code such as a UMLS Concept Unique Identifier (CUI) as is done in rules-based IE systems.

Instead, we used a mixture of reference sources to expand *label aliases* while preserving the original term. If an alias for a term is found rather than the term itself, the input as classified as a positive sample for that label. For medications, we linked our labels to Canadian brand names provided by *drugbank.ca* [Wishart *et al.*, 2017]. For event and condition label we used the open-access and collaborative consumer health vocabulary [University of Utah, Biomedical Informatics Department, 2011].

Given the volume of research in biomedical fields, these represent only a small sampling of quality label alias sources, though many sources replicate similar information (e.g., brand names of medications can be found in numerous sources). We limited our expansion to these two sources after experimenting with additional sources such as the primary US medication reference, *RxNorm* and the *SNOMED CT* browser. We found that an injudicious addition of labels resulted in a rapid increase in false positive rates compared to the GSC labels as well as markedly slowing run-times.

The final step in pre-processing is to convert our text labels into *multi-hot* vectors

using `scikit-learn's` MultiLabelBinarizer. This works by building an n length vector from the training data where n is the total number of labels in this category (Condition, Event or Medication). For a single record's label vector $\hat{Y}$, $y_i = \{0, 1\}$, $Y = [y_1, y_2, \ldots, y_n]$ and if C_Y is the count of labels for this record then $\sum_{i=0}^{n} y_i = C_Y$. For example, given the possible labels of {blue, green, red}, a multihot vector for a record with the labels blue and green would be [1,1,0] while a record labelled green and red would be [0,1,1].

After this pre-processing, we observe that the majority of labels have only one positive sample in our training set. This issue is addressed in the section *Data Expansion*.

Chapter 4

Feature Extraction

4.1 Word Embeddings

In order to optimize training with our ML algorithms, it's necessary to reduce document text to a more compact numerical representation. In this project, we consider the term frequency-inverse document frequency (`tf-idf`) methodology to convert words to vectors. A `tf-idf` vector is constructed by fitting all of our document text to `scikit-learn's` implementation and selecting the most informative 1000 features.

Models were also trained using a 100-dimension `word2vec` embedding [Mikolov *et al.*, 2013] that we created from the MIMIC III and IMPACT labels after removing common English stop words. The word embedding was created using negative sub-sampling skip-gram and we experimented with window sizes of $[3, 5, 7]$ as well as minimum word counts of $[1, 3, 5]$. All embeddings performed relatively the same with window size having no perceptible impact; however optimal performance was achieved with a minimum count of three, likely reflecting the removal of *hapax legomena* consisting largely of misspellings and poor OCR data in the MIMIC III corpus. Text is converted into features through a *mean embedding vector* for individual word vectors. If a word is not found in the embedding, we assign a 0 value vector.

By being trained on over 40,000 clinical records and the exact labels created by the annotators, the vectors more accurately capture semantic similarities in our GSC for terms that don't occur in our training data, without including errors from the IMPACT corpus itself. After the embedding parameters above, our MIMIC vectors have 152,000 unique words. For comparison, the project's GSC consisted of approximately 7 million tokens with a vocabulary of between 85,000 and 102,000 words for `Clean` and `Raw` data respectively which still includes noise (merged words, misspellings >2 edit distance).

4.2 POS Tags

As well as converting text into a numeric format via embeddings, we generate a normalized vector for each input of Part of Speech (POS) tags. Tags are generated via the Natural Language Toolkit (NLTK) [Bird *et al.*, 2009]. Text is tokenized and it is here that our `Clean` OCR input is split into sentences. The tokens are then passed to NLTK's tokenizer which assigns each token one of the Penn Treebank POS tags. For each record or sentence, we count the occurrence of each token to create a vector of 36 integers, $X = [x_1, x_2, \ldots, x_n]$, where n is the POS label.

As shown earlier, there is a large degree of variance in the number of tokens for each sample so we normalize the vector as follows:

$$\vec{X} = \frac{1}{m}[x_1, x_2, \ldots, x_n]$$

where

$$m = \sum_{i=0}^{n} x_i$$

Unfortunately, most conventional linguistic analysis does little to improve overall classification efforts. We speculate that this is due to the inherent noise in the data quality as well as the style of clinical notes reflecting a mix of incomplete sentences and abbreviations. Our data reports nearly 25% proper nouns (NNP) and 15% nouns (NN). These rates vary, depending on the purpose of other texts, so a good comparison would be clinical notes such as those in the MIMIC set. In MIMIC 17% of tokens are NNP and 18% are NN which is a total of 5% less than seen in IMPACT data and over 30% fewer proper nouns to nouns. It is likely that the tagger is confusing unknown terms, including many nouns, with names and tagging them as proper nouns.

Chapter 5

Data Expansion

5.1 Training Data Balance

Earlier work on CAMH data highlighted the importance of training data size and the ratio of positive to negative samples to a model's predictive power. One of the biggest findings in earlier CAMH IE research was that using a small, unbalanced training set of 87% negative samples, ML algorithms would quickly overfit and develop a bias towards the negative class [Geraci *et al.*, 2017]. An alternative was to train the model on a smaller, but balanced data set.

The expectation, then, is that models trained on our full data sets are at risk of considering most new records as negative samples given that for most labels, more than 97% of the training data is negative. However, simply balancing the data set by balancing IMPACT positive and negative samples will cause a second issue, similar to that in the research above. Due to the limited size of the training set, using an even smaller subset increases the model's volatility by making overfitting more likely.

We adapt this approach for each label in our GSC, since we can treat each individual label as a boolean classification. For each label we split the training set of 100 records into positive and negative samples. We then supplement the samples with as few positive samples from the MIMIC III corpus as needed to balance the samples. MIMIC documents are identified through an n-gram search of the raw clinical notes for the label string.

Even with this simplistic search, we were able to drastically reduce the number of labels with only a single sample. This was important as, per our test protocol, the first positive sample from the IMPACT data is held back as test data. Prior to this supplement, between 55% and 78% of our GSC would not be classifiable due to having only negative training samples — after supplementing, this was reduced to 34% to 44%. *Figure 5.1* illustrates how supplementing with MIMIC samples increases label frequency.

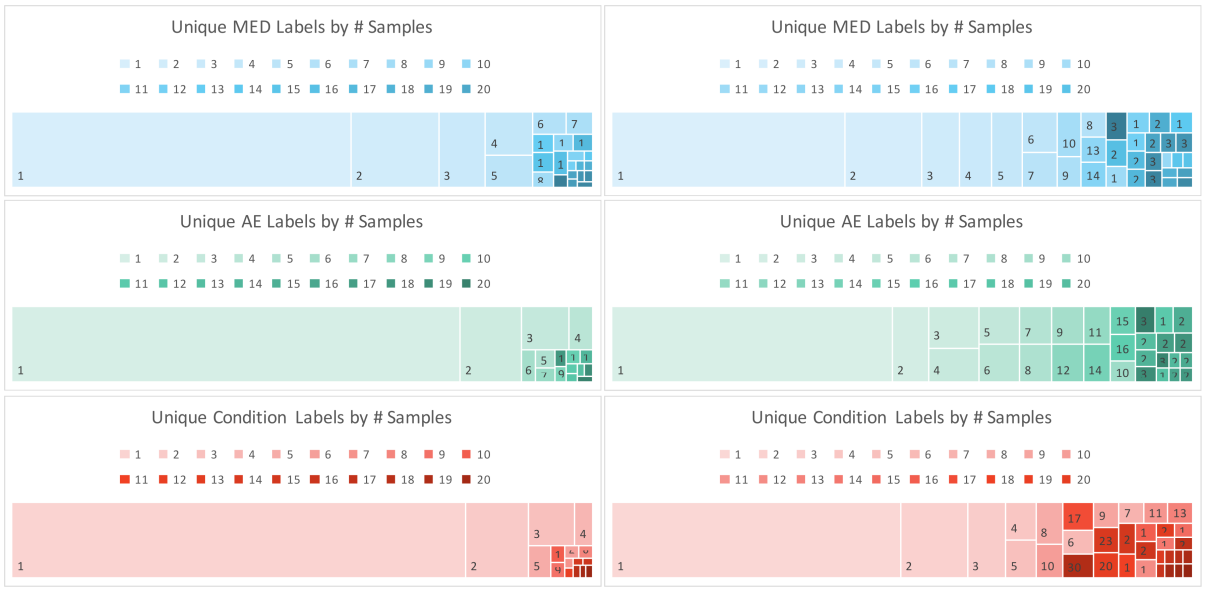


Figure 5.1: Counts of documents that were annotated with a specific label. Counts range from 1 to 117 documents with that label. The IMPACT only data set is shown on the left; the mixed data set including MIMIC is on the right.

We only use a limited number of positive samples for each label due to the existing data imbalance in IMPACT data. The introduction of additional negative samples would increase the amount of *sample selection bias* in our models [Huang *et al.*, 2007]. By creating a mixed training sample by obfuscating the data source from our classifiers we can augment training sets, as well as reducing class imbalance.

We experimented with multiple methods to select supplemental samples, including minimizing the euclidean distance between our samples when they are projected into the same embedding space. These trials almost immediately replicated our overfitting issues as the MIMIC samples acted as copies of the limited IMPACT samples. Models fitted this way performed extremely poorly on any held-out samples, except in certain situations where the number of IMPACT samples was close to the same as the supplemental samples that we added. In these cases, the supplemental data did not appear to have any effect.

We determined that a random subsampling of supplemental documents was the most reliable method for boosting performance, though its impact was primarily in allowing us to train classifiers on labels that only had one sample. The classifiers were incorrect the majority of the time, but there were incremental gains.

5.2 Synthentic Resampling

A common step in improving accuracy in ML is to resample the training data to prevent overfitting to the majority. Resampling methods involve either *oversampling* the minority classes to increase their weight during training or *undersampling* the majority classes to reduce their weight.

Synthetic Minority Oversampling [Chawla *et al.*, 2002] introduces new training points based on re-sampling training data. Adaptive synthetic sampling [He *et al.*, 2008], another *oversampling* technique, uses an iterative approach based on the class-wise accuracy on the testing set to determine what synthetic data to add to training. In terms of *undersampling*, prototype generation via Cluster Centroids attempts to replace clusters of majority samples with $n_{minority}$ average prototypes where n is the total number of training samples. The final resampling approach is Random Undersampling which performs prototype selection by extracting a subset of size $n_{minority}$ from $X_{majority}$.

5.3 Record to Sentence Labelling

Another promising avenue of research was to *bootstrap* the limited but long training samples into lists of labelled sentences. In this way, our ML algorithms would have more data to analyze and we can apply more conventional regularization via the algorithms. The primary risk with this approach is that the GSC labels were applied at the record level without indication of specific sentences matching specific labels.

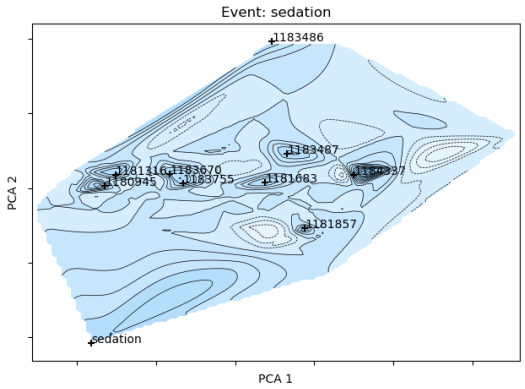
We converted the binary labels y on a document with m sentences as follows:

$$\text{Given binary vector: } \vec{Y} = [y_1, y_2, \ldots, y_n] | \{y_i \in \{0, 1\}\}$$
$$\text{Create real vector: } \vec{\hat{Y}} = \frac{1}{m}[\hat{y_1}, \hat{y_2}, \ldots, \hat{y_n}] | \{\hat{y_i} \in \Re | 0 \leq \hat{y} \leq 1\}$$

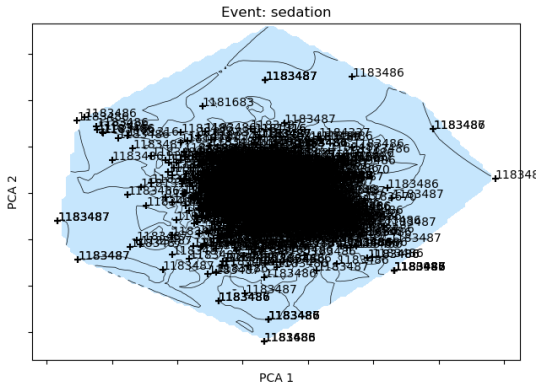
We then applied a second pass to sentences in the document. If a sentence contained an exact match for label j, we set $\hat{y}_j = 1$. We did not zero out any vectors as there is no indication that other sentences did not inform the label assigned by the human annotator. However, further research could show a way to dynamically adjust label vectors based on distance measurements of sentences to specific labels when projected into a shared vector space, similar to how word embeddings are used in current state-of-the-art analogical reasoning [Pennington *et al.*, 2014].

Figure 5.2 shows several examples of this process projected into the shared embedding space that has been reduced via Principal Component Analyses to two dimensions. Examples are from each category. Contours are darker around positive examples to give

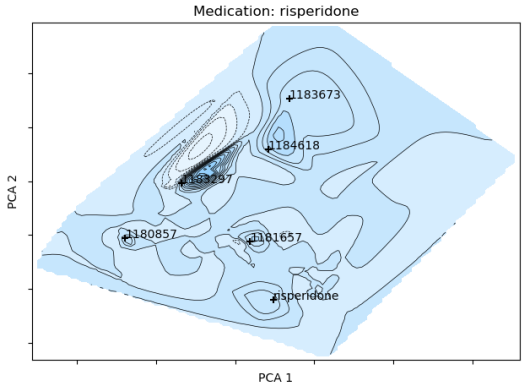
a visual indication of the "terrain" that our algorithms will operate in. Note that the principal components and embedding space are the same for each graph; the difference in layout is indicative of the interpolation of points used to create the contour with white space indicating that no samples lay in that area. It is clear that adding the sentence information adds significant noise to each classification.



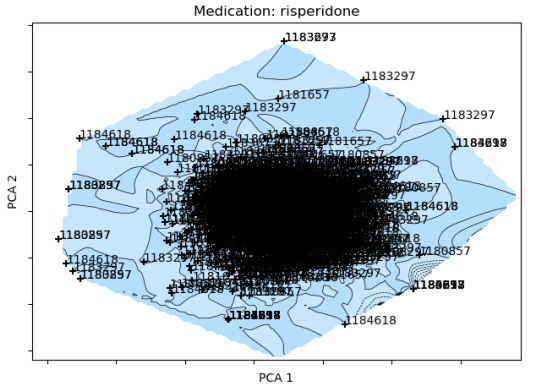
(a) Event: Sedation documents



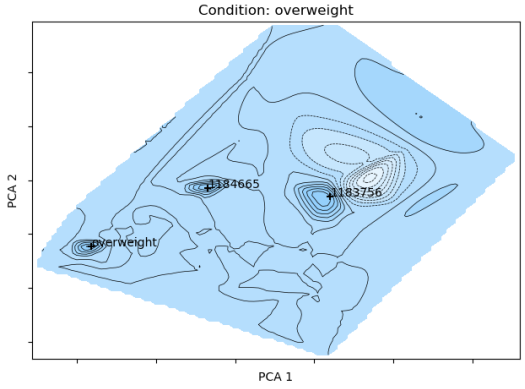
(b) Event: Sedation sentences



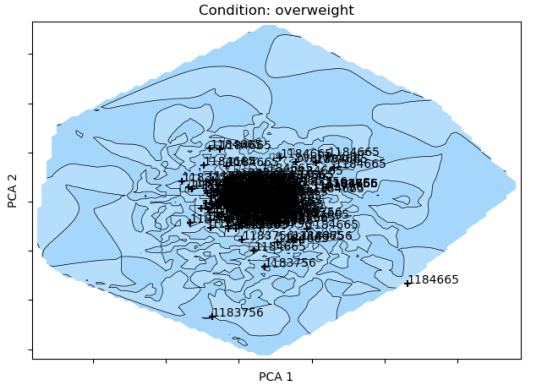
(c) Medication: Risperidone documents



(d) Medication: Risperidone sentences



(e) Condition: Overweight documents



(f) Condition: Overweight sentences

Figure 5.2: Contour plots of documents, sentences, and label terms projected into a shared embedding space, where greater saturation / elevation indicates the presence of a positive label for that input. Samples are drawn from labels with between 2 and 9 annotated records, as this is representative of the majority of labels. Figures on the left are for document and label term embeddings; figures on the right are for sentence and label term embeddings for the same records.

Chapter 6

Algorithm Selection

6.1 Training Segmentation

Model parameters, as well as word embedding types were evaluated via hyper-parameter optimization using `scikit-learn's GridSearchCV` model selection class and 5-fold cross-validation of MIMIC supplemented training data. Details on the optimal parameters can be found in *Table 6.1*.

For each unique label in a category (Condition, Event, and Medication), we trained and evaluated the 10 models as binary classifiers. Only labels where the number of positive training samples ≥ 1 were considered — for each category, Condition, Event and Medication, this equated to 391, 359 and 413 unique labels and classifiers respectively:

(1) A Multinomial Naive Bayes (**MNB**) classifier serves as the baseline for the project.

(2) A Logistic Regression (**LR**) classifier had shown the most success in a recent study on classifying treatments for patients with *major depressive disorder*, which is one of the positive keywords annotators used on this project, so it was evaluated [Perlis, 2013].

(3) A Random Forest (**RF**) classifier is often a successful approach in NLP competitions and has also been used in multiple modules in the open-source, Mayo Clinic's cTAKES, although this latter relies on significant pruning and *a priori* expert knowledge to achieve its consistent successes [Savova *et al.*, 2010].

(4-5) Two versions of a multi-layer perceptron network were evaluated as previous research similar models to achieve their best results. The difference between the models is primarily in how they were optimized, with one using Gradient Descent (**GMP**) and the other Stochastic Gradient Descent (**SMP**). Model hyper-parameters were configured independently and resulted in 3 layers with 100 neurons/layer as opposed to the previous

Table 6.1: GridSearchCV results showing the hyper-parameter ranges that were tested and the best values for each classifier in bold.

	Model
1	MNB Alpha = [1e-09, 1e-06, 1e-03, **0.001**, 0.01, 0.1, 0.5, 1.0] Fit_Prior = **True** / False n-gram Range = [**1**, 2, 3]
2	LR Alpha = [1e-09, 1e-06, **1e-03**, 0.001, 0.01, 0.1, 0.5, 1.0]
3	RF Max_Features = **auto**, sqrt, log2 Samples_Split = [2, 4, **8**] N_Estimators = [5, 10, 20, 30, **50**, 100] n-gram Range = [1, 2, **3**]
4	GMP Hidden_layers = [(50,50,50), **(100,100,100)**, (200,200,200)]
5	SGP Alpha = [1e-09, 1e-06, **1e-03**, 0.001, 0.01, 0.1, 0.5, 1.0] n-gram Range = [1, 2, **3**]
6	SVM Alpha = [1e-09, 1e-06, 1e-03, **0.001**, 0.01, 0.1, 0.5, 1.0] Class weight = **None**, balanced Classifier Penalty = **None**, L2, L1, elasticnet n-gram Range = [**1**, 2, 3]
7	SVC C = [0.001, 1.0, **10**, 25, 50, 100] Gamma = [0.001, 1.0, **10**, 25, 50, 100] Kernel = linear, **rbf** Class weight = None, **balanced** n-gram Range = [**1**, 2, 3]
8	CNN Optimizer = SGD, RMSProp, Adagrad, Adadelta, Adam, **Adamax**, Nadam Batch_Size = [10, **20**, 50, 80, 100] Epochs = [**5**, 10]
9	LSTM Batch_Size = [**10**, 20, 50, 80, 100] Epochs = [**5**, 10]
10	biLSTM Batch_Size = [**10**, 20, 50, 80, 100] Epochs = [5, **10**]

work's 200 neurons/layer.

(6-7) Similarly, two different Support Vector Machines were evaluated as these are some of the best performing NLP classifiers [Sarikaya *et al.*, 2014]. The implementations differ in optimization method as well as kernel type, with one model using a `linear` kernel (**SVM**) and the other using a `rbf` kernel (**SVC**).

(8) A convolutional neural network (**CNN**) was evaluated due to its significantly faster training speed compared to more-complex RNN models and often comparable results [Zhang & Wallace, 2017].

(9-10) Both a simple long short-term memory RNN (**LSTM**) [Hochreiter & Schmidhuber, 1997], and bidirectional LSTM RNN (**biLSTM**) [Graves & Schmidhuber, 2005] with only a single LSTM layer were evaluated based on their current status as the state-of-the-art in NLP tasks. In order to facilitate training times, these networks also include an initial convolutional layer to reduce the number of features subsequent layers had to evaluate, which resulted in a 500% increase in training speed with nearly identical accuracy.

Note that multi-label classification algorithms were also explored, however with over 1500 unique labels for 100 training samples, classifiers ended up either overfitting or underfitting by predicting all one value or the other.

6.2 Evaluation

This project's primary goal was to reduce the volume of documents requiring manual review by hospital staff. Operationally, this is done by only reviewing documents that our classified as positive matches for a given label. To assess the performance of different classifiers against this goal we not only evaluate the results' overall **Accuracy**, a common measure of model performance, but also the **Positive Predictive Value** (PPV).

PPV is the same as a model's *Precision* and indicates the proportion of true positives that a model predicts. This gives us an indication of how successful this classifier would be operationally as a low PPV would result in additional, needless manual review. PPV is calculated as follows:

$$PPV = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} = \frac{\text{True Positives}}{\text{Predicted Positives}}$$

Accuracy is a poor assessment of quality in this project as, with over 97% of most

training data representing the negative class, a model can score highly simply by always predicting negative results. It can, however, serve as a secondary goal in our study should any models exhibit similar PPV. We calculate it as follows:

$$Accuracy = \frac{\sum TruePositive + \sum TrueNegative}{\sum Total}$$

Scores are calculated for each classifier and each label term on the specific **17** record subsets held out as test samples for that label term. Results are then averaged across the test records and all label terms in a category to create three PPV and *Accuracy* scores per classifier, one for each category.

If the classifier was trained on sentence-level source data, as discussed in section **??**, we need to convert sentence-level predictions to record-level predictions as our labels are at the record-level. To accomplish this, we apply a threshold t to the label-wise sum of sentence level prediction vectors Y_s for all n sentences in a given record r. This lets us calculate the record-level prediction vector $\hat{Y_r}$ of length m predictions where m is equal to the number of unique labels. To maximize PPV we set $t = 0$ and apply it as follows:

$$\text{Given } r = [s_1, s_2, \ldots, s_n]$$

$$\text{and } Y_s = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m]$$

$$\text{then } Y_r = [y_1, y_2, \ldots, y_m]$$

$$\text{such that } y_m \begin{cases} 1, & \text{if } \sum_{i=1}^{n} \hat{y}_m > t \\ 0, & \text{otherwise} \end{cases}$$

Chapter 7

Results

7.1 Comparison of Results

To begin, note that a simple unigram search outperformed the cTAKES baseline. This is a combination of naturalistic labels, noisy source data that can make a rule-based IE system compound initial errors, and the baseline US lexicon that the software uses. All these factors are mitigated through our NLP pipeline. However, from a user's perspective this may be a reason clinical IE systems are not more common; if a search function through your OS can generate reasonably accurate results quickly, there is little motivation to add complications unless there is significant gain.

The best-performing models on the `Clean` data set were a *rbf* Support Vector Machine and a Gradient Descent Perceptron. In fact, most classifiers tend to cluster around a 55% **ACC** an 80% **PPV**. Examining further it appears the most classsifiers score high on PPV by flagging a lot of False Positives; since we are not reporting recall, these numbers are only reflected in overall accuracy, which is little better than a random coin flip.

However, spot-checking results, it may also be the case that the original annotations are incorrect. Our models are able to identify labels that have agreement with the cTAKES rule-based engine. This implies that, for this task, the models are capable of replicating clinical IE heuristics with minimal supervision.

Secondly, we note that using a sentence-level data set increased the *Positive Predictive* value of all models, even if it resulted in a higher false positive rate. The results presented in *Table 7.1* are for classifiers trained on this data set and a scatter plot of these results is shwon in *Figure 7.1*

Thirdly, we note that the false positives rate increased the more label aliases we apply. This may actually reflect correct classifications of the documents that were missed by human annotators. Thus, the use of structured reference information to support

naturalistic searches is an area where we can focus more effort to support *human-in-the-loop* classification. Even if our evaluation scores are lower, this is still meaningful data for the medical researchers.

It also appears that there is a strong correlation between *Condition* and *Event* labels. This makes sense, as several of the labels are duplicated between these categories. In order to train a finer grained classifier, additional information on why the GSC differentiated between these labels will be needed.

We also note that though our word embeddings were trained from 2,000,000 clinical notes, they contained only 150,000 terms. They could benefit from being trained on additional sentences. Our current embeddings return vectors for only 50% of the tokens used in IMPACT data, whereas word embeddings trained on PubMed data have over 2,200,000 terms and return vectors for 80% of tokens [Moen & Ananiadou, 2013]. Theoretically, the more matches returned by an embedding results in more precise vectors and greater separation of data points after converting tokens to embeddings. This is because unmatched tokens are assigned the same 0-value vector which 'squashes' different text to the same point in our embedding space.

Although the PubMed vectors matched more tokens, applying them resulted in decreased performance of all classifiers. We speculate that the poor performance is because several of the label terms don't occur in the PubMed vectors. This is likely due to these tokens appearing infrequently in the corpus used to train these embeddings (e.g., the Canadian-branded medication *Cipralex*). This means classifiers "miss" exact-term matches and underperform compared to the simpler approach of n-gram matching.

Ideally, a next step would be to merge PubMed vectors with our self-trained embeddings to maximize the number of matched vectors. Unfortunately, PubMed vectors were released in an uneditable format.

A surprise in the results is how little impact POS tags had on classification. We speculate this is due to poor quality source data "confusing" the tagger. When we removed the POS vectors from training runs, there were no noticeable differences in cross-validation results. We continued to append the vectors in the final results.

In terms of synthetic data, all classifiers benefited from *undersampling* techniques with Random Undersampling consistently boosting results. In contrast, the *oversampling* techniques only increased model overfitting as the oversampled points remained close to existing training data and did not help the models' predictions on held out test data.

Table 7.1: Best 12 classifier results for the three different IMPACT categories.

	PPV %	**Accuracy %**	**Average %**
Condition			
Exact	27	**97**	62
cTAKES	14	88	51
MNB	66	54	60
LR	58	58	58
RF	58	58	58
GMP	**86**	57	**71**
SMP	73	53	63
SVM	71	55	63
SVC	**85**	55	**70**
CNN	81	52	67
LSTM	78	52	65
biLSTM	79	56	68
Event			
Exact	14	97	55
cTAKES	16	**98**	57
MNB	69	56	62
LR	49	59	54
RF	49	59	54
GMP	**81**	56	**68**
SMP	73	56	64
SVM	67	57	62
SVC	**79**	58	**69**
CNN	75	56	66
LSTM	69	57	63
biLSTM	71	52	62
Medication			
Exact	48	**97**	**73**
cTAKES	23	59	41
MNB	76	56	66
LR	46	42	44
RF	73	55	64
GMP	**92**	56	**74**
SMP	76	51	63
SVM	65	59	62
SVC	**88**	58	**73**
CNN	81	53	67
LSTM	75	55	65
biLSTM	77	59	68

Figure 7.1: Accuracy vs. Precision for Label Categories and Classifiers

7.2 Limitations

The biggest constraint on this project, and many practical clinical IE tasks, is the limited number of samples. ML algorithms perform best when trained on millions of records and this GSC consisted of only 100 training documents. This meant it is easy for models to 'memorize' the data and overfitting happens early even when regularization parameters are included, supplemental data is added and synthetic samples are used.

Our attempt to turn the low quantity, but lengthy, training documents into sentence training sets had limited success likely due to the inheritance of the same record labels by every sentence in a record. Turning binary labels into *fractional* labels uniformly across most sentences acted as a magnification of noise when the sentences were projected into an embedding space. Instead of a single point representing a positive value, *fractional* labels added many more points and the nature of the records is that they cover an extensive time period so the assumption that a sentence at the start of the document will have similar semantic information to one near the end does not hold. In essence, this technique acted as an extreme regularizer by introducing additional noise to our training set.

The root of most of the issues in our project is poor quality source data. In general, text pre-processing and noise reduction were unable to overcome OCR artifacts which, similar to rule-based IE systems and the `deid` package, caused a chain effect of missed and incorrect labelling.

As a final note, our GSC has only a single annotator per record. As evidenced by label spelling mistakes and exact matches of some label n-grams to records labelled as a negative sample, there is a significant chance that the GSC labels don't reflect the complete ground truth. Additional annotators per record or guidance on any additional criteria used by the team would allow us to replicate the guidance in our pipeline as well as calculate an inter-annotator agreement score as part of the training pipeline.

Chapter 8

Discussion and Further Work

The goal of this project was to reduce the amount of manual data review required by expensive research staff. It highlights that state-of-the-art ML methodologies can yield significant benefits to the current process, especially when considering critical real-world constraints such as limited and noisy source data, plain-language search terms, data privacy, missed annotations, and the complexity of current clinical IE tools. It is unlikely that a highly accurate model will ever be trained on this or similar data sets without significant pre-processing or leveraging larger medical data sets. However, a practical follow-up task is to create an ensemble of the best-performing models. This would allow researchers to quickly query EHR data and receive a restricted subset of highly probable documents for review which can then be manually annotated as part of the study protocol.

Data pre-processing is another area where a more comprehensive approach could be researched. The project showed the flaws in the existing de-identification protocol where some personal names were not found or were mis-identified. In fact, no automated de-identification software is 100% accurate; this is a significant hurdle to transfer learning using clinical data, due to the strict limits around protected health information. Similar to suggestions made by Friedman *et al.* [2013] we propose continuing to develop a powerful, user-friendly IE tool that can be used by research staff who are already authorized to view this sensitive information as opposed to cloud or server-based solutions.

Our research also leads to some interesting observations that could be expanded on in further research. First, it appears that when our GSC is projected into a word embedding space, medication names exhibit an interesting form of clustering. Instead of a brand name being near to its generic name, brand names cluster with other brand names and generic names cluster together as well. Could these word embedding distance observations be used as a way to track the appearance and use of new drugs in different jurisdictions?

Second, the approach of including label terms in the same embedding space boosted performance of the classifiers. However, without the terms occurring in a sentence, we noticed that all labels appear to cluster in one area of the embeddings, whereas documents with the same term are distributed much more evenly (see *Figure 5.2*). It should be possible to use templates or a medical language model to generate more realistic sentences for these labels prior to embedding them. Taking a more generative approach to supplemental data could provide an interesting avenue for further research.

Third, in contrast to a rules-based IE system, our pipeline's results can be continually improved through unsupervised and semi-supervised methods, rather than relying on the manual addition of new rules. For example, in order for cTAKES to recognize Canadian-branded medications, users must download and reformat a dictionary of medications into XML and then update the configuration files of the application; this level of computer programming becomes increasingly rarer the smaller a research team.

Our pipeline can be continually improved by non-programmers in several ways. First, training better word embeddings is a reliable, unsupervised method for improving this pipeline. Users can save relevant raw text files and the pipeline will perform the ingestion, pre-processing, and updates of the embedding model. Second, by using limited supervision (e.g., including only positive samples from external sources) these corpi can be used to augment custom lexicons or correct class imbalances via transfer learning.

The release of open, relevant data sets is only increasing. The NLP pipeline approach explored in this paper provides a means to leverage and encourage this trend. The data sets we applied were initially developed for other clinical IE research projects and released after onerous, manual de-identification to obscure PHI was performed.

We believe that the simpler it is for users to implement a clinical IE tool, the greater the likelihood that it will be deployed by smaller teams. Allowing teams to assess the level of PHI remaining visible after processing encourages the release of relevant data for other users either internally (if PHI remains visible) or to external researchers. This leads to a virtuous cycle where users incrementally improve the performance of their clinical IE pipelines by sharing restricted, de-identified, or public data sets between projects. By encouraging this cycle, we posit that machine learning algorithms, whose performance increases in proportion to the volume of relevant training data, will increase in effectiveness at clinical IE through increased adoption of this approach.

Bibliography

Bird, Steven, Klein, Ewan, & Loper, Edward. 2009. *Natural language processing with Python: Analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Bodenreider, Olivier, & McCray, Alexa T. 2003. Exploring semantic groups through visual approaches. *Journal of Biomedical Informatics*, **36**(6), 414–432.

Callard, Felicity, Broadbent, Matthew, Denis, Mike, Hotopf, Matthew, Soncul, Murat, Wykes, Til, Lovestone, Simon, & Stewart, Robert. 2014. Developing a new model for patient recruitment in mental health services: a cohort study using Electronic Health Records. *BMJ Open*, **4**(12), e005654.

Cavoukian, Ann. 2011. *Privacy by design in law, policy and practice*. `https://www.ipc.on.ca`. [Online; accessed October-2018].

Chawla, Nitesh V, Bowyer, Kevin W, Hall, Lawrence O, & Kegelmeyer, W Philip. 2002. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, **16**, 321–357.

Church, Kenneth W, & Hovy, Eduard H. 1993. Good applications for crummy machine translation. *Machine Translation*, **8**(4), 239–258.

Fink, Eugene, Kokku, Princeton K, Nikiforou, Savvas, Hall, Lawrence O, Goldgof, Dmitry B, & Krischer, Jeffrey P. 2004. Selection of patients for clinical trials: an interactive web-based system. *Artificial Intelligence in Medicine*, **31**(3), 241–254.

Fivez, Pieter, Šuster, Simon, & Daelemans, Walter. 2017. Unsupervised context-sensitive spelling correction of clinical free-text with word and character n-gram embedding. *Pages 143–148 of: 16th Workshop on Biomedical Natural Language Processing of the Association for Computational Linguistics*.

Friedman, Carol, Rindflesch, Thomas C, & Corn, Milton. 2013. 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.

Gardner, James, & Xiong, Li. 2009. An integrated framework for de-identifying unstructured medical data. *Data & Knowledge Engineering*, **68**(12), 1441–1451.

Geraci, Joseph, Wilansky, Pamela, de Luca, Vincenzo, Roy, Anvesh, Kennedy, James L, & Strauss, John. 2017. Applying deep neural networks to unstructured text notes in electronic medical records for phenotyping youth depression. *Evidence-based Mental Health*, **20**(3), 83–87.

Gonzalez-Hernandez, G, Sarker, A, OConnor, K, & Savova, G. 2017. Capturing the Patients Perspective: a Review of Advances in Natural Language Processing of Health-Related Text. *Yearbook of Medical Informatics*, **26**(01), 214–227.

Graves, Alex, & Schmidhuber, Jürgen. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, **18**(5), 602–610.

He, Haibo, Bai, Yang, Garcia, Edwardo A, & Li, Shutao. 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *Pages 1322–1328 of: Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*. IEEE.

Herbert, Deanna, Neves-Pereira, Maria, Baidya, Ruth, Cheema, Sheraz, Groleau, Sarah, Shahmirian, Anashe, Tiwari, Arun K, Zai, Clement C, King, Nicole, Müller, Daniel J, *et al.*. 2018. Genetic testing as a supporting tool in prescribing psychiatric medication: Design and protocol of the IMPACT study. *Journal of Psychiatric Research*, **96**, 265–272.

Hochreiter, Sepp, & Schmidhuber, Jürgen. 1997. Long short-term memory. *Neural Computation*, **9**(8), 1735–1780.

Huang, Jiayuan, Gretton, Arthur, Borgwardt, Karsten M, Schölkopf, Bernhard, & Smola, Alex J. 2007. Correcting sample selection bias by unlabeled data. *Pages 601–608 of: Advances in neural information processing systems*.

Johnson, Alistair EW, Pollard, Tom J, Shen, Lu, Li-wei, H Lehman, Feng, Mengling, Ghassemi, Mohammad, Moody, Benjamin, Szolovits, Peter, Celi, Leo Anthony, &

Mark, Roger G. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data*, **3**, 160035.

Legislative Assembly of the Province of Ontario. 2000 (Nov). *Bill 159 - The Personal Health Information Privacy Act.* `http://www.health.gov.on.ca/en/common/ministry/publications/reports/phipa/bill_159.pdf`. [Online; accessed November-2018].

Malik, Majid Bashir, Ghazi, M Asger, & Ali, Rashid. 2012. Privacy preserving data mining techniques: current scenario and future prospects. *Pages 26–32 of: Computer and Communication Technology (ICCCT), 2012 Third International Conference on.* IEEE.

Maloney, Chris, Sequeira, Ed, Kelly, Christopher, Orris, Rebecca, & Beck, Jeffrey. 2013 (Dec). *PubMed Central.* 2013 Nov 14 [Updated 2013 Dec 5]. In: The NCBI Handbook [Internet]. 2nd edition. Bethesda (MD): National Center for Biotechnology Information (US); 2013-. Available from: `https://www.ncbi.nlm.nih.gov/books/NBK153388/`.

Marks, Lara, & Power, Emmett. 2002. Using technology to address recruitment issues in the clinical trial process. *Trends in Biotechnology*, **20**(3), 105–109.

McDonald, Keltie C, Bulloch, Andrew GM, Duffy, Anne, Bresee, Lauren, Williams, Jeanne VA, Lavorato, Dina H, & Patten, Scott B. 2015. Prevalence of bipolar I and II disorder in Canada. *The Canadian Journal of Psychiatry*, **60**(3), 151–156.

Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg S, & Dean, Jeff. 2013. Distributed representations of words and phrases and their compositionality. *Pages 3111–3119 of: Advances in Neural Information Processing Systems.*

Moen, SPFGH, & Ananiadou, Tapio Salakoski2 Sophia. 2013. Distributional semantics resources for biomedical text processing. *Pages 39–43 of: Proceedings of the 5th International Symposium on Languages in Biology and Medicine, Tokyo, Japan.*

Neamatullah, Ishna, Douglass, Margaret M, Li-wei, H Lehman, Reisner, Andrew, Villarroel, Mauricio, Long, William J, Szolovits, Peter, Moody, George B, Mark, Roger G, & Clifford, Gari D. 2008. Automated de-identification of free-text medical records. *BMC Medical Informatics and Decision Making*, **8**(1), 32.

Norvig, Peter. 2007. *How to write a spelling corrector.* `http://norvig.com/spell-correct.html`. [Online, accessed October-2018].

Ogren, Philip V, Savova, Guergana K, Chute, Christopher G, *et al.*. 2007. Constructing evaluation corpora for automated clinical named entity recognition. *Page 2325 of: Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics; Building Sustainable Health Systems*. IOS Press.

Olieman, Alex, Beelen, Kaspar, van Lange, Milan, Kamps, Jaap, & Marx, Maarten. 2017. Good Applications for Crummy Entity Linkers: The Case of Corpus Selection in Digital Humanities. *Pages 81–88 of: Proceedings of the 13th International Conference on Semantic Systems*. ACM.

Pennington, Jeffrey, Socher, Richard, & Manning, Christopher. 2014. Glove: Global vectors for word representation. *Pages 1532–1543 of: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*.

Perlis, RH, Iosifescu, DV, Castro, VM, Murphy, SN, Gainer, VS, Minnier, Jessica, Cai, T, Goryachev, S, Zeng, Q, Gallagher, PJ, *et al.*. 2012. Using electronic medical records to enable large-scale studies in psychiatry: treatment resistant depression as a model. *Psychological Medicine*, **42**(1), 41–50.

Perlis, Roy H. 2013. A clinical risk stratification tool for predicting treatment resistance in major depressive disorder. *Biological Psychiatry*, **74**(1), 7–14.

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

Ross, Sue, Grant, Adrian, Counsell, Carl, Gillespie, William, Russell, Ian, & Prescott, Robin. 1999. Barriers to participation in randomised controlled trials: a systematic review. *Journal of Clinical Epidemiology*, **52**(12), 1143–1156.

Santorini, Beatrice. 1990. Part-of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision, 2nd printing).

Sarikaya, Ruhi, Hinton, Geoffrey E, & Deoras, Anoop. 2014. Application of deep belief networks for natural language understanding. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, **22**(4), 778–784.

Savova, Guergana K, Masanz, James J, Ogren, Philip V, Zheng, Jiaping, Sohn, Sunghwan, Kipper-Schuler, Karin C, & Chute, Christopher G. 2010. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association*, **17**(5), 507–513.

Smith, Ray. 2007. An overview of the Tesseract OCR engine. *Pages 629–633 of: Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, vol. 2. IEEE.

University of Utah, Biomedical Informatics Department. 2011 (May). *Open Access, Collaborative Consumer Health Vocabulary Initiative*. `http://consumerhealthvocab.org/`. [Online - accessed September-2018].

US Department of Health and Human Services. 2018 (Oct.). *Guidance on HIPAA & Cloud Computing*. `https://www.hhs.gov/hipaa/for-professionals/special-topics/cloud-computing/index.html`. [Online; accessed October-2018].

Vellido, Alfredo, Martín-Guerrero, José David, & Lisboa, Paulo JG. 2012. Making machine learning models interpretable. *Pages 163–172 of: ESANN*, vol. 12. Citeseer.

Wang, Yanshan, Wang, Liwei, Rastegar-Mojarad, Majid, Moon, Sungrim, Shen, Feichen, Afzal, Naveed, Liu, Sijia, Zeng, Yuqun, Mehrabi, Saeed, Sohn, Sunghwan, *et al.*. 2017. Clinical information extraction applications: a literature review. *Journal of Biomedical Informatics*.

Wishart, David S, Feunang, Yannick D, Guo, An C, Lo, Elvis J, Marcu, Ana, Grant, Jason R, Sajed, Tanvir, Johnson, Daniel, Li, Carin, Sayeeda, Zinat, *et al.*. 2017. DrugBank 5.0: a major update to the DrugBank database for 2018. *Nucleic acids research*, **46**(D1), D1074–D1082.

Zhang, Ye, & Wallace, Byron. 2017. A Sensitivity Analysis of (and Practitioners Guide to) Convolutional Neural Networks for Sentence Classification. *Pages 253–263 of: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1.