Extracting Patient Events from Clinical Text Using Natural Language Processing

by Shanika Rajapakshe

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

B.SC. COMPUTER SCIENCE HONOURS

in

Irving K. Barber School of Arts and Sciences

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA
(Okanagan)

April 2017

ABSTRACT:

The clinical progress note is one of the most crucial documents in medicine. Details on a patient's well being, progression through treatment, and future care management are noted by the attending physician. This makes the progress note a dense source of clinically relevant information. However, it requires a significant amount of time for humans to read these notes and determine the best course of action for the patient. Event Extraction, the recognition and mining of events from text data, offers a streamlined method to accomplish this task. Traditionally, computers are not well equipped to deal with the unstructured nature of human language. Improvements in machine learning and natural language processing in the recent past have lead to a proliferation of open source tools to deal with the nuances of natural language.

This paper illustrates a natural language pipeline utilizing these tools for intelligent event extraction in a clinical setting, dealing specifically with clinical progress notes of lung cancer patients.

Table of Contents

Acknowledgements

I would like to thank Dr. Jonn Wu, senior radiation oncologist and Chair of the Provincial Head and Neck Tumour Group at the BC Cancer Agency for the motivation and idea behind this project, Dr. Cheryl Ho for access to the patient cohort, which allowed for the pipeline to be created and trained, and Dr. Ramon Lawrence for his advice and guidance in creating this project.

List of Tables

List of Figures

Chapter 1: Introduction

The electronic patient chart is a standard document used in medicine for storing information about a patient's care. Due to the complexity of healthcare data, the patient chart is a source of both structured and unstructured information. Extracting meaningful insights from this data is a laborious process. Currently, the data must be understood by humans with relevant domain knowledge for this information to be obtained. The Cancer Agency Information System (CAIS) is the main repository of patient records for the British Columbia Cancer Agency (BCCA). Details such as lab reports and physicians' notes are stored as quantitative data or free text. In the current workflow, health care professionals are required to open each patient's relevant information from a desktop computer to properly identify important patient events. Although methods have been suggested in improving the access of this data, it is still a difficult process to navigate CAIS to obtain these events.

Due to recent improvements in the fields of computer and data science, the field of machine learning (ML) has matured significantly over the past 10 years. Natural language processing (NLP), a subfield of machine learning, has also seen tremendous growth in the past decade. Recent events, such as IBM's Watson winning Jeopardy in 2011 and the popularity of Google's search engine have shown the potential of these new technologies. In addition, the World Health Organization (WHO) has suggested that eHealth, or the incorporation of information and communication technologies for health as an area of important growth in healthcare.

The combination of these factors has allowed for an influx in ML and NLP research in the field of healthcare informatics. However, many of these studies are performed by combinations of well-endowed research universities (Mount Sinai, University of California), large hospital systems (Memorial Sloan Kettering, Veterans Health Administration) and established information technology companies (IBM, Microsoft). These groups are able to circumnavigate the two major roadblocks of ML in medicine, lack of expertise and lack of relevant medical datasets.

At this point, NLP can be performed by those with access to the correct data and training. The critical point is gathering the necessary amount of accurate data to make a machine learning solution viable. Due to the nature of ML, large amounts of meaningful data must be leveraged in order to train and test the eventual machine learner. In addition, this data must be prepared in a way that would allow it to be easily ingested by the ML system. The information supplied by CAIS has the potential of being such a data set.

This thesis will explore the steps involved in the creation of a NLP pipeline to perform automatic data extraction from an open text document stored in CAIS.

Chapter 2

Background

2.1 : Key Terms and Definitions

BC Cancer Agency (BCCA)

The organization provides the entire spectrum of cancer care to the BC population. This includes cancer prevention, treatment, and awareness.

CAIS

The BCCA's electronic health record (EHR), which stores patient information from across their cancer care, including progress notes, images, lab reports and blood work.

Machine Learning (ML)

A subfield of computer science and statistics involved in 'teaching' computers to make decisions on data provided for them.

Supervised Learning

A machine learning paradigm characterized by providing pre-labeled data from which the algorithms 'learn' from.

Natural Language Processing (NLP)

A subfield of machine learning and linguistics, involving the use of computers in understanding and working with human language.

Personally Identifiable Information (PII)

Information that can uniquely identify an individual, such as full name, address, date of birth, and telephone number.

Cancer

Cancer is a set of related diseases characterized by abnormal cell growth. In almost all forms of cancer, body cells begin to proliferate and divide in an uncontrolled manner, and eventually spread to other tissues if left unchecked. As the disease progresses, masses of these cells form growths known as tumours.

Clinically relevant events

For the purpose of this paper, these are defined as events in a patient's care that have major implications to all future points of care. In the context of lung cancer, clinically relevant events are:

- Therapies
  - Surgery
  - Chemotherapy
  - Radiation
- Recurrence
- Palliative Care

Cancer Therapies are methods used to treat cancer. Most often, they are involved in the degradation or removal of the cancerous tissues. Additionally, it may also involve targeting tissues that are at risk for developing cancer in the future. These at risk tissues can be noted due to their location relative to the affected tissue, or physical characteristics that make them more susceptible.

Surgery

The most common method of dealing with cancer is surgery. Generally, this is the removal of the cancerous tissues. In the context of lung cancer, there are several kinds of surgeries, most notably:

- Pneumonectomy – removal of entire lung
- Lobectomy – removal of the entire lobe (lungs made of 5 lobes, 3 on right, 2 on left)
- Segmentecomy or wedge resection – part of a lobe is removed
- Pleurectomy – removal of the pleura, linings surrounding the lungs
- Sleeve Resection – used for treating cancers in large airways of the lungs

Chemotherapy

Chemotherapy is the use of powerful drugs to treat cancer. These drugs are specially designed to specifically target the quickly growing cancerous cells. Chemotherapy often carries side effects, including but not limited to nausea, hair loss and lowered immune system function. Specific chemotherapy drugs relevant to lung cancer include:

- Cisplatin
- Vinorelbine
- Gefitinib (Iressa)
- Carboplatin
- Gemcitabine1
- Docetaxel
- Erlotinib (Tarceva)
- Anastrozole

Radiation

Using high energy radiation to treat cancer by shrinking tumours and killing cancer cells. Radiation damages cancer cells by damaging the cells DNA, causing them to break apart and be digested by the body's natural processes. Type of radiation used in cancer therapy include X-Rays, Gamma Rays and charged particles.

Palliative Care

This refers to any treatment that is focused on pain management rather than curing the patient. It can be any of the three types of treatment listed above. Typically, it is used in end of life scenarios.

Recurrence

A cancer that is found after treatment and after a period of time when the cancer could not be detected. Generally, any new cancer developed by the patient in the vicinity of their previous cancer is considered a recurrence. However, a cancer recurrence can also occur further away from the original tumour site.

- Local Recurrence – the new cancer has come back in the same place as the original
- Regional Recurrence – the new cancer has come back in lymph nodes close to the original cancer
- Distant Recurrence – the new cancer has come back in a different part of the body, some distance away from the original site.

2.2: Overview

Healthcare creates a multitude of data. Before the introduction of information systems, most of this data was either lost or remained locked in paper records. Now with the move towards more digital systems, most of this data is captured and stored electronically. However, much of it is still difficult to use at scale. This is due to the unstructured nature of most healthcare data. Unlike structured data, those that traditionally fit within a relational database, unstructured data is much more difficult for a computer system to manage. This is mainly due to the ambiguities within the unstructured data. Text and multimedia, such as audio and video, are the main forms of unstructured data.

In order to continue improving the healthcare system, it is important to utilize this data. It offers the potential of uncovering large scale effects and population-level trends within the healthcare system, and understanding the health of the population at an intimate level. As more data is added, the value of easy access and understanding of the data will also improve, since the tools that use the data will only improve as more relevant data is added.

Additionally, this deluge of data creates another problem; that of information overload. Work needs to go in to minimize physician exposure to extraneous data. However, due to the complex nature of healthcare information, it becomes very difficult to untangle what is important, and what is not relative to a patient's care.

The Problem

With all of this in mind, using unstructured data in a healthcare setting can be complicated. Thus, we thought it would be beneficial to explore a small sub-problem within this context. This became the task of performing event extraction from unstructured clinical

text. Specifically, extracting relevant events from clinical progress notes of lung cancer patients.

The clinical progress note is a document dictated by a physician to record a patient's care at a point in time. Typically, information such as the physical and mental well being, medication being taken, treatments planned or undergoing, and relevant medical history are noted. Thus, it is a very information dense document. In order for an attending physician to get an understanding of a patient, these clinical progress notes need to be retrieved and reviewed. This can turn into a laborious process, especially in the case of chronic health conditions such as cancer, which have patients repeatedly visiting healthcare centres over the course of several months or years. With such a large amount of information to keep track of, as well as the seriousness of content, it is important to examine this information.

However, there are times when physicians wish to review certain key events in a patient's care. Many of these events, such as treatments or cancer recurrence, drastically effect the patient's care going forward, and are the most important for physicians to review. Yet physicians still need to review all progress notes to find all instances of these events for any given patient. Extracting these events from clinical progress notes would work towards streamlining this process for physicians. The hope is for them to quickly find the notes relevant for a given event, allowing for less time looking at computers and more time caring for the patient.

2.3: The Data

The dataset is the set of clinical progress notes from all patients referred to the BCCA from January 2005 to December 2010 with surgically resected stage II non-small-cell lung carcinoma. This resulted in a set of 261 unique patients with 2668 clinical progress notes.

2.4: Event Extraction

Event Extraction is the extraction of complex combinations of relations between entities, performed after a series of initial NLP. (Kaymak, de Jong, Caron, Hogenboom, & Frasincar, 2016). In more common terminology, it is the act of recognizing and retrieving information about events from text data. It utilizes knowledge from several fields, including computer science, linguistics, artificial intelligence, and statistics.

One of the major difficulties with Event Extraction is due to the ambiguous nature of unstructured text. Many sentences that appear logically similar have drastically different meaning depending on context. This in turn makes it very difficult for event extraction systems to perform at the level of humans.
For example:

Let's eat, Grandma.

Let's eat Grandma.

The above sentences mean two very different things due to the inclusion of a single comma. Such changes in meaning are common throughout the English language.

Thanks to its interdisciplinary nature, event extraction has grown include several approaches, which can be summarized as three distinct methods.

Expert Systems

These utilize rules or patterns to perform event extraction. The strength of these systems is that the rules are easy for humans to understand (i.e. People make the rules which the computer follow). However, these are inflexible due to these very same rules. To add new events to the system, entirely new rules need to be defined, and then applied to the text.

Data Driven Systems

These approaches utilize machine learning and statistical methods. In these methods, the textual information is transformed into different representations that can be applied to algorithms and mathematical models in a step known as training. These trained models can then be applied to new instances of text to perform the event extraction.

Unlike expert systems, it is often very difficult to understand the reasoning behind the decisions these algorithms make. Thus, they are treated as black box methods, where the internal mechanics of the resulting trained algorithm are not understood by humans.

Hybrid Systems

Using one of the two methods above exclusively leads to many different problems. A popular approach to solving these problems is using a hybrid approach. This approach uses methods from both expert and data driven systems to utilize the strengths of each while minimizing the weaknesses. For example, one can bootstrap expert systems with machine learning, or optimize data driven systems using sentence patterns.

These systems require more data than expert systems, as they do involve some machine learning. However, they require much less data than purely data driven systems.

Chapter 3

Dr. Zed

Dr. Zed is the NLP pipeline and resulting trained machine learning tool for data driven event extraction from clinical text.

There are three major components of the system:
- Data Preparation
- Feature Extraction
- Algorithms.

Dr. Zed utilizes a data driven approach to event extraction. All of the software is written in Python 3.5, and is open source with additional open source python software libraries.

The libraries are as follows:
- Scipy
- Numpy
- Pandas
- NLTK
- SciKit-Learn
- Faker

3.1: Data Preparation

This step refers to manipulating and altering the original dataset into a format digestible for the subsequent parts of the system. The original dataset was spread across several patients in the BCCA's CAIS system. Additionally, the progress notes were saved as PDF, a filetype with large overhead. Several steps were required to extract the data from CAIS and modify it into a manageable format.

3.1.1: Reading in the charts

CAIS holds extremely sensitive patient information. These include date of birth sex, location and other personally identifiable information. Due to this, CAIS only allows access to the system through the user front end, and has no allowances for third party apps or programs to access to the data. Manual effort was required to download the patient progress notes.

Physicians, after meeting with a patient, call in to a dedicated number to verbally dictate notes about the patient. These are then automatically transcribed via software or manually transcribed by a trained transcriptionist.

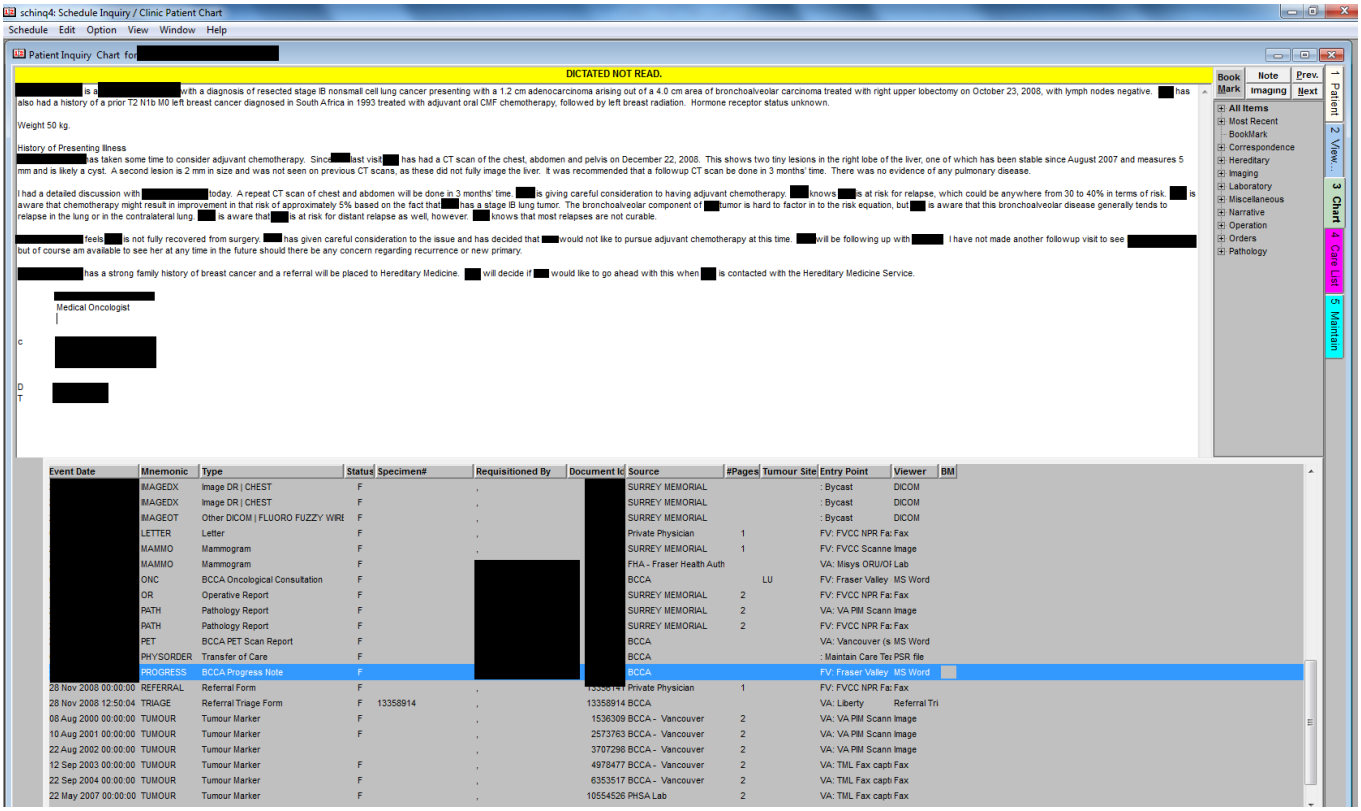
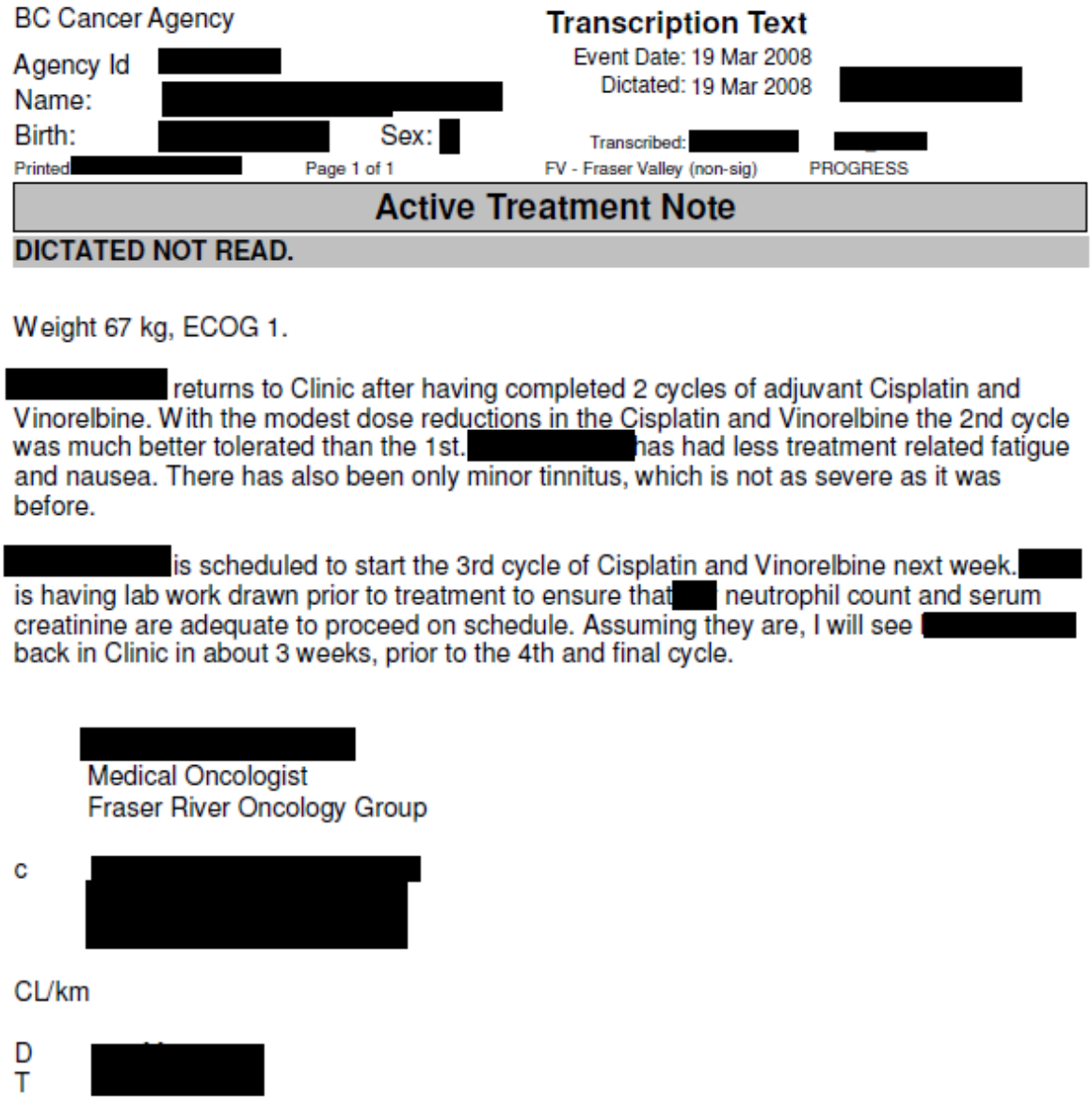


Figure 1: Screenshot of CAIS

Most clinical progress notes were stored as PDF files containing the dictated information as well as some header information regarding the date of the note, date of transcription, and identity of the physician. Some notes were uploaded in different file formats, most notably pictures and screenshots of physical notes saved as jpeg images. Optical Character Recognition was proposed, to read these images. These screenshots added up to a very small set of notes (less than 50), and the workload required to use these images was judged as too high for this small amount of additional data.



BC Cancer Agency

Agency Id [redacted]
Name: [redacted]
Birth: [redacted] Sex: [redacted]
Printed [redacted] Page 1 of 1

Transcription Text
Event Date: 19 Mar 2008
Dictated: 19 Mar 2008 [redacted]
Transcribed: [redacted] [redacted]
FV - Fraser Valley (non-sig) PROGRESS

Active Treatment Note

DICTATED NOT READ.

Weight 67 kg, ECOG 1.

[redacted] returns to Clinic after having completed 2 cycles of adjuvant Cisplatin and Vinorelbine. With the modest dose reductions in the Cisplatin and Vinorelbine the 2nd cycle was much better tolerated than the 1st. [redacted] has had less treatment related fatigue and nausea. There has also been only minor tinnitus, which is not as severe as it was before.

[redacted] is scheduled to start the 3rd cycle of Cisplatin and Vinorelbine next week. [redacted] is having lab work drawn prior to treatment to ensure that [redacted] neutrophil count and serum creatinine are adequate to proceed on schedule. Assuming they are, I will see [redacted] back in Clinic in about 3 weeks, prior to the 4th and final cycle.

[redacted]
Medical Oncologist
Fraser River Oncology Group

c [redacted]

CL/km

D [redacted]
T

Figure 2: Example of a clinical progress note

The downloaded PDF files were then stored securely on a dedicated desktop machine located at the BCCA in Kelowna, BC within the PHSA firewall.

3.1.2: Conversion from PDF to .txt

Although PDF files are a standard method of storing textual data, they have additional information that is in addition to the actual text. Therefore, it was necessary to convert the PDF files into .txt files for ease of use within the proposed pipeline.

The open source Xpdf library and its pdftotext command line application was used to convert the PDF files to .txt files. The resulting .txt files were stored in the same manner as the PDF files. On some occasions, adjacent words were combined but for the most part the conversion worked very well. These infrequent errors were left in the finished text documents, as they did not compromise the validity of the documents, and it would have been a time-consuming process to fix. Dealing with these errors is a small price to pay, as it was much easier to use the script than to manually copy and paste text from each of the 2668 pdfs into separate .txt files.

3.1.3: Anonymization

As stated previously, clinical progress notes contain personally identifiable information. It was imperative to remove this information in order to protect the privacy of the patients involved. The clinical progress note has some standardization regarding PII of the patients. In the header of the document, the name of the patient, ID number (referenced by the BCCA to uniquely identify the patient) date of birth, and sex of the patient are all indicated.

A python script was written to recognize these PII, and generate fake data to replace them. However, as a way to retrieve specific patients in the future, a dictionary was maintained to link the fake name and ID to the real patient. This dictionary was also kept on the dedicated desktop at the BCCA in Kelowna, BC for security reasons.

3.1.4: Sentence Tokenization

Sentence Tokenization refers to the splitting of a document into sentences. There are several methods to do this, usually involving Regex expressions. Both the NLTK and SpaCy libraries were explored for this component. However, NLTK outperformed the SpaCy implementation, and was consequently used in the final pipeline.

3.1.5: Word Tokenization

Similar to sentence tokenization, word tokenization splits a sentence into individual words. This is an easier task than sentence tokenization as spaces define separation between words in most English sentences. Again, both NLTK and SpaCy were proposed for this step. To maintain consistency with the sentence tokenization, NLTK was used in the final pipeline.

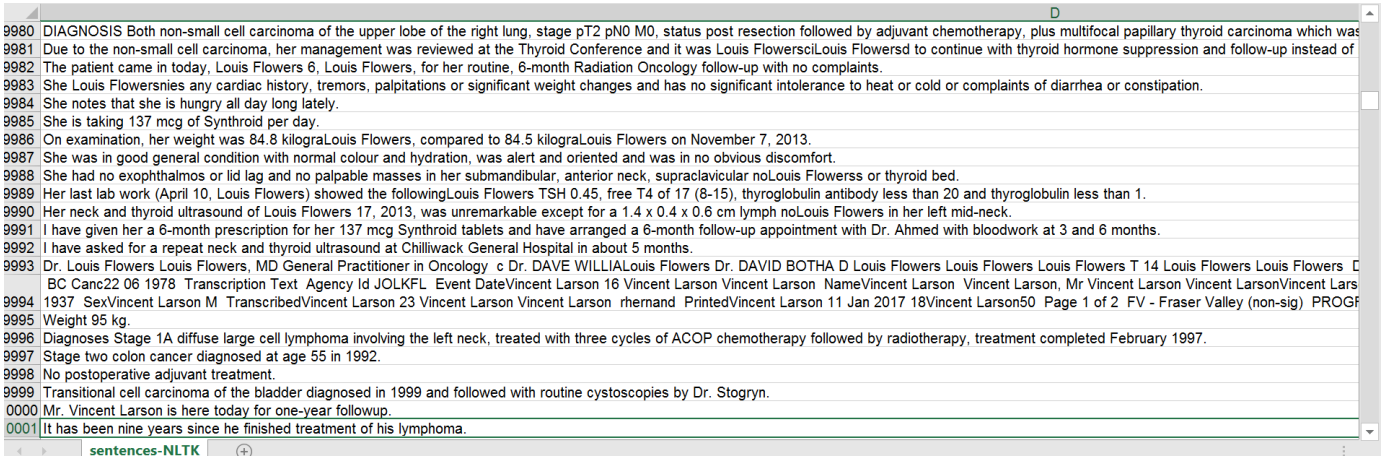
3.1.6: Stemming

Even in small corpa of documents, many words are present. Often, there are a significant number of words that are structurally related to other words. A good example of this are different tenses of verbs. A method of simplifying downstream parts of the pipeline involves reducing the vocabulary size by trimming words to their word stem. Stemming is one such process, where inflected and derived words are reduced to their word stems. In this case, broad rules are applied, which remove morphological affixes from words.

The Snowball and Porter stemmers are two commonly used stemmers, and both have implementations in NLTK. After some testing, the Snowball stemmer was chosen for use in the pipeline.

3.1.7: Categorizing Each Sentence

In order to get a good training and test set, the presence or absence of each event needed to be coded manually. To do this, the extracted sentences were ported to an excel sheet. The source document title, sentence number, and manual coding for all events was recorded for each sentence.



	D
9980	DIAGNOSIS Both non-small cell carcinoma of the upper lobe of the right lung, stage pT2 pN0 M0, status post resection followed by adjuvant chemotherapy, plus multifocal papillary thyroid carcinoma which was
9981	Due to the non-small cell carcinoma, her management was reviewed at the Thyroid Conference and it was Louis FlowersciLouis Flowersd to continue with thyroid hormone suppression and follow-up instead of
9982	The patient came in today, Louis Flowers 6, Louis Flowers, for her routine, 6-month Radiation Oncology follow-up with no complaints.
9983	She Louis Flowersnies any cardiac history, tremors, palpitations or significant weight changes and has no significant intolerance to heat or cold or complaints of diarrhea or constipation.
9984	She notes that she is hungry all day long lately.
9985	She is taking 137 mcg of Synthroid per day.
9986	On examination, her weight was 84.8 kilograLouis Flowers, compared to 84.5 kilograLouis Flowers on November 7, 2013.
9987	She was in good general condition with normal colour and hydration, was alert and oriented and was in no obvious discomfort.
9988	She had no exophthalmos or lid lag and no palpable masses in her submandibular, anterior neck, supraclavicular noLouis Flowerss or thyroid bed.
9989	Her last lab work (April 10, Louis Flowers) showed the followingLouis Flowers TSH 0.45, free T4 of 17 (8-15), thyroglobulin antibody less than 20 and thyroglobulin less than 1.
9990	Her neck and thyroid ultrasound of Louis Flowers 17, 2013, was unremarkable except for a 1.4 x 0.4 x 0.6 cm lymph noLouis Flowers in her left mid-neck.
9991	I have given her a 6-month prescription for her 137 mcg Synthroid tablets and have arranged a 6-month follow-up appointment with Dr. Ahmed with bloodwork at 3 and 6 months.
9992	I have asked for a repeat neck and thyroid ultrasound at Chilliwack General Hospital in about 5 months.
9993	Dr. Louis Flowers Louis Flowers, MD General Practitioner in Oncology c Dr. DAVE WILLIALouis Flowers Dr. DAVID BOTHA D Louis Flowers Louis Flowers Louis Flowers T 14 Louis Flowers Louis Flowers D
	BC Canc22 06 1978 Transcription Text Agency Id JOLKFL Event DateVincent Larson 16 Vincent Larson Vincent Larson NameVincent Larson Vincent Larson, Mr Vincent Larson Vincent LarsonVincent Lars
9994	1937 SexVincent Larson M TranscribedVincent Larson 23 Vincent Larson Vincent Larson rhernand PrintedVincent Larson 11 Jan 2017 18Vincent Larson50 Page 1 of 2 FV - Fraser Valley (non-sig) PROGF
9995	Weight 95 kg.
9996	Diagnoses Stage 1A diffuse large cell lymphoma involving the left neck, treated with three cycles of ACOP chemotherapy followed by radiotherapy, treatment completed February 1997.
9997	Stage two colon cancer diagnosed at age 55 in 1992.
9998	No postoperative adjuvant treatment.
9999	Transitional cell carcinoma of the bladder diagnosed in 1999 and followed with routine cystoscopies by Dr. Stogryn.
0000	Mr. Vincent Larson is here today for one-year followup.
0001	It has been nine years since he finished treatment of his lymphoma.

sentences-NLTK

Figure 3: The sentences from all progress notes

At the time of the writing of this thesis, 10000 of the total 55679 sentences were coded. This was deemed as an appropriate size for testing the rest of the pipeline components.

	E	F	G	H	I
1	Recurrence	Chemotherapy	Radiation	Surgery	Palliative
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	1	0	0	0
6	0	0	0	0	0
7	0	0	0	0	0
8	0	0	0	0	0
9	0	0	0	0	0
10	0	0	0	0	0
11	0	0	0	0	0
12	0	0	0	0	0
13	0	0	0	0	0
14	0	0	0	0	0
15	0	0	0	0	0
16	0	0	0	0	0
17	0	0	0	0	0
18	0	0	0	0	0
19	0	0	0	0	0
20	0	0	0	0	0
21	0	0	0	0	0

sentences-NLTK

Figure 4: The event categorization for sentences

3.2: Feature Extraction

3.2.1: Bag of Words (BOW)

The bag of words approach is a staple of natural language processing.
In this method, a vocabulary is defined. Typically, it is the set of all words found across all documents. In the case of Dr. Zed, this would refer to all of the words across all patient charts in the dataset. Each sentence is represented as a vector of length equal to the size of the vocabulary. Each entry in the vector corresponds to one of the words in the vocabulary. The value stored at each vector bin refers to the number of times that word appeared in the sentence. Small word vectors can also visually represent this information about text, as they can be represented as histograms.

BOWs make a few broad assumptions about the relationships between words. It assumes that the occurrence of a word is independent of the occurrence of other terms. This makes modelling far easier, but does go against our view of sentences, as there is implicit understanding that certain words predispose other words to show or not show up in the rest of the sentence.

Most documents will only contain a small subsection of the total set of words from all documents. Thus, the vectors representing each sentence will be sparse i.e. Will contain

many zeros. For example, a set of 10000 short documents such as emails will use roughly 100000 words total, but each document will only contain 100 to 1000 unique words.

BOWs are very versatile, and have been utilized for a variety of other tasks. These include image categorization (Li, Tao, & Xian-Sheng, 2011) and time series data (She, Nahavandia, Kouzani, Wang, & Liu, 2013).

The classical BOW approach has a few weaknesses. By representing sentences as vectors, it loses the ordered nature of text. As natural language relies heavily on word order to express meaning, a lot of very important information is lost.

Improvements

Tf-IDF Weighting

In large sets of documents, words that occur frequently (ex. the, a, but, and) do not hold much meaningful information about specific sentences. In the traditional BOW method, which uses direct counts of words, these pervasive words would minimize the effect of more interesting but less frequent words. Thus, it can be considered more important to weigh words that occur less frequently more heavily than words that occur more frequently. The Tf-IDF method manages this problem by transforming word counts into floating point values. This reweighting is performed according to the following equation:

$$TFIDF(t,d) = tf(t,d) \times idf(t)$$

Where Term Frequency (TF) refers to the number of times a term (in this case, a word) occurs in a document (which in this case is a sentence).

The IDF is defined as

$$idf(t) = \log\frac{1+n_d}{1+df(d,t)} + 1$$

N_d is the total number of documents, df(d,t) is the number of documents that contain the term t

These vectors are then normalized by the Euclidian Norm

$$v_{norm} = \frac{v}{\|v\|^2} = \frac{v}{\sqrt{{v_1}^2+{v_2}^2+\cdots+{v_n}^2}}$$

The resulting floating point values are then used in the vector.

N-Gram Models

This approach tackles the loss of word ordering in the traditional BOW by including n-grams rather than words. An n-gram is a sequence of continuous n items from a given portion of text. N-grams of individual words are called unigrams. The vocabulary of an n-gram model consists of all n-grams within the set documents.

Table 1: N-Grams for the phrase 'to be or not to be'

N-Gram	Unigram (1-Gram)	Bigram(2-Gram)	Trigram (3-Gram)
	to, be, or, not, to, be	to be, be or, or not, not to, to be	to be or, be or not, or not to, not to be

Implementation

The Scikit-Learn Library contains a very effective BOW representation as the `CountVectorizer` class. This can represent any N-gram representation of information. Additionally, TF-IDF weighting is also implemented through the related `TfidfVectorizer` class.

3.2.2: Parts of Speech (POS) Tagging

POS tagging is the process of denoting each word in a sentence with its corresponding part of speech. These refer to grammatical parts of the sentences, such as nouns, verbs, adjectives, etc.

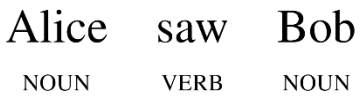


Figure 5: Sentence with its corresponding POS Tags

The most used POS representation is coded using the Penn Treebank.

Table 2: Subset of the Penn Treebank POS tags (Bird, Klein, & Loper, 2014)

Tag	Meaning	English Examples
ADJ	adjective	*new, good, high, special, big, local*
ADP	adposition	*on, of, at, with, by, into, under*
ADV	adverb	*really, already, still, early, now*
CONJ	conjunction	*and, or, but, if, while, although*
DET	determiner, article	*the, a, some, most, every, no, which*
NOUN	noun	*year, home, costs, time, Africa*
NUM	numeral	*twenty-four, fourth, 1991, 14:24*
PRT	particle	*at, on, out, over per, that, up, with*
PRON	pronoun	*he, their, her, its, my, I, us*
VERB	verb	*is, say, told, given, playing, would*
.	punctuation marks	*. , ; !*
X	other	*ersatz, esprit, dunno, gr8, univeristy*

Many words have several meanings depending on context. This makes POS Tagging a challenging task. Despite this difficulty, many algorithms have been proposed for POS Tagging.

Like Event Extraction methods, POS Tagging algorithms fall into three groups: rule based, data driven, and hybrid approaches. Additionally, many groups supply their own pretrained data driven POS Taggers. Although these are more specific to the domain they were trained upon, they work very well for most POS tasks. Thus, a pretrained data driven POS Tagger was utilized for this pipeline.

Training of our own POS Tagger was also explored, but eventually rejected due to the effort required to manually code the POS tags for all sentences.

Implementation

Both the SpaCy and NLTK libraries have prebuilt POS taggers. Dr. Zed used NLTK's pretrained POS Tagger, the `maxent_treebank_pos_tagger`.

3.2.4: Dependency Parse Trees

Dependency refers to the idea that words are connected by directed links. Typically these are centered around the head of the sentence, most often the tensed verb. The Dependents, all other parts of the sentence, are either connected directly to the sentence head or are through a path of dependencies. Dependency Parse Trees are structures that illustrate these relationships as a directed graph.

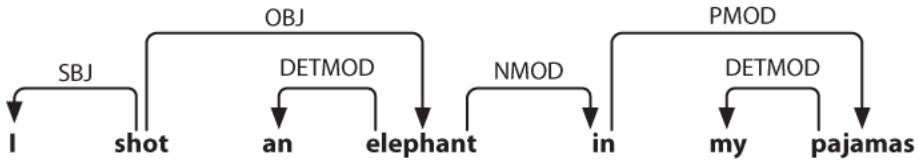


Figure 6: Example of a Dependency Parse Tree (Bird, Klein, & Loper, 2014)

Implementation

In NLTK, POS Tags are used as inputs for obtaining the Dependency Parse Trees.

3.2.5: Named Entity Detection

Named entities are proper noun phrases that refer to a specific individual, organization or entity.

Table 3: Examples of Named Entities

NE type	Examples
ORGANIZATION	*Georgia-Pacific Corp., WHO*
PERSON	*Eddy Bonte, President Obama*
LOCATION	*Murray River, Mount Everest*
DATE	*June, 2008-06-29*
TIME	*two fifty a m, 1:30 p.m.*
MONEY	*175 million Canadian Dollars, GBP 10.40*
PERCENT	*twenty pct, 18.75 %*
FACILITY	*Washington Monument, Stonehenge*
GPE	*South East Asia, Midlothian*

This step uses pre-trained models to determine which words or phrases refer to named entities, or proper nouns. The NLTK library contains a pretrained model, which was used for the pipeline.

3.2.6: Overall Pipeline

The image below represents the overall pipeline. The black box represents the coded results for each sentence. White boxes signify the feature representations of the text. Grey boxes represent raw data from the source documents.

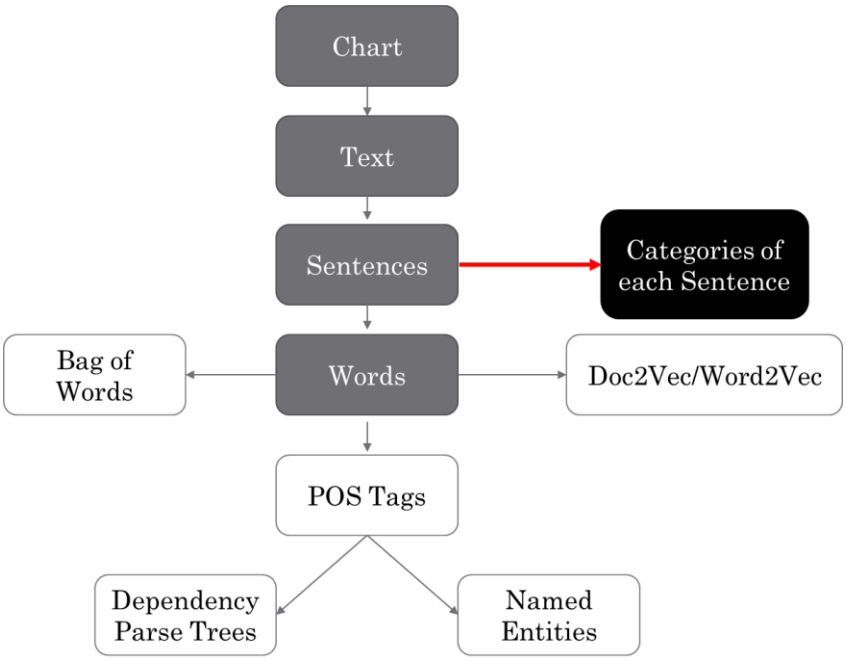


Figure 7: Overall Pipeline

Chapter 4

Algorithms and Methods

As stated previously, much of the pipeline is used in creating the inputs for the machine learning portion. Pipeline:

- Bag of Words (BOW)
- N-Grams
- TF-IDF Normalization
- POS Tags
- Dependency Parse Trees
- Named Entities

Due to the volume of information, BOWs, N-grams BOWs and TF-IDF BOWs were to be used as inputs for the following algorithms. Future work would consider applying the other inputs and potentially combining them with the BOW representations.

All the algorithms used below were implemented by the Scikit-Learn Library. Each was trained by 6000 sentences and their corresponding event codings, with the remaining 4000 held out as the test set.

4.1: Logistic Regression

Logistic regression is a method of classification utilizing a specialized form of linear regression, which itself is a method of supervised learning.

Background

Linear regression is a method of supervised learning, which is simple and powerful. The traditional linear regression model is as follows:

$$E[y] = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

Where E[y] is the expected value of the response variable Y that we are trying to predict, $\beta_0, \ldots, \beta_p$ are the coefficients for the linear equation, and $X_1, \ldots, X_p$ are the values of the p predictor variables for that observation.

However, this method is not appropriate for predicting binary or categorical values. (James, Witten, Hastie, & Tibshirani, 2014) However, a transformation can be applied to work around this. In this case, we are modelling p(X), the probability that X is of a certain category. We start with the logistic function:

$$p(X) = \frac{e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}{1 + e^{\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p}}$$

This can be manipulated into the following:

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}$$

Taking the log of both sides leads to

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

The left-hand side is a quantity known as the log-odds, or the logit, thus

$$logit(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

That is, the logistic regression model has a logit that is linear in X (James, Witten, Hastie, & Tibshirani, 2014).

Multi-level classifiers

Traditionally, logistic regression is used for binary classification. However, as some of the data has more than one class (such as recurrence) it is necessary to allow for this. An approach to this is the One-Versus-Rest (OVR) approach. A classifier is created for each category, with members of that category coded as positive examples, and members of any other category coded as negatives examples. The logistic regression performed by Scikit-Learn supports this approach to multi-level classification, as well as several others.

4.2: Nearest Neighbor Classifier

This is a straightforward approach which uses 'similar' observations to classify new instances. Upon exposure to the training set, the nearest neighbor algorithms store the location of each observation, making it an example of instance based learning. When new observations are given, they are classified by a majority vote system. Of the K nearest neighbors to the observation in question, the category with the most neighbors will be the classification of the new observation.

Basic nearest neighbors algorithms use uniform weights, which weighs the contribution from each neighbor equally. Alternative methods can weigh neighbors more if they are closer to the observation in question.

4.3: Multi-Layer Perceptron

These are basic implementations of neural networks, which are composed of layers of input and output nodes. It is composed of several layers of nodes, called Perceptrons. Each

perceptron acts as a 'neuron' and takes in the summation of a number of inputs, and applies an activation function. It then goes on to forward this resulting value, either to another set of nodes in the following layer, or as a final output. Multilayer Perceptrons can use a variety of activation functions, including the logistic function and sigmoid function.

Additionally, the input, hidden and output layers can have differing numbers of nodes.

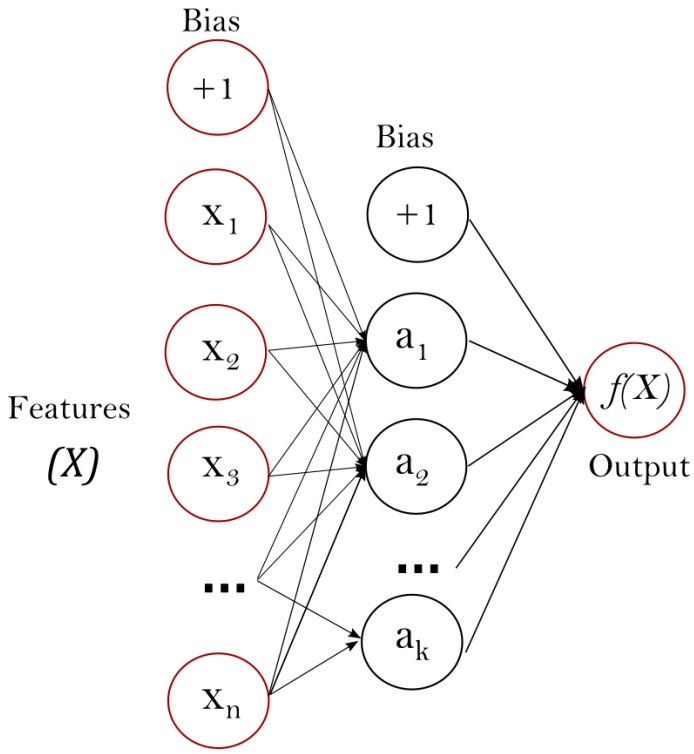


Figure 8: Architecture of a typical multilayer perceptron

Because of their complexity, multilayer Perceptrons and other neural networks are able to deal with messy data, especially those that may contain mistakes. This makes it a useful algorithm for this project, as there is a possibility that some labelled sentences may be labelled incorrectly. However, these algorithms are slow to train, and require a lot of data in order to be useful.

4.4: Other Algorithms

The algorithms above were the method with the most interesting results. Other algorithms utilized included Gaussian Naïve Bayes Classifiers, Support Vector Machines, decision trees and linear discriminant analysis.

Chapter 5

Evaluation

To evaluate which combinations of features and algorithms performed the best, a standard set of evaluation metrics must be defined. In natural language processing, and event extraction these metrics are precision, recall, and the F1-Score.

Precision

Also known as positive predictive value, is the fraction of correctly classified sentences (true positives) over the total number of sentences classified as that category (true positives plus false positives).

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Recall

This is the sensitivity, or fraction of correctly classified sentences over the true total of sentences that are of that category.

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

F1-Score

This is hybrid approach to measuring precision and recall. It weighs each, and uses that score to keep both in mind when measuring the performance of the system. The F1-Score is defined as the following:

$$F1\ Score = \ 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Additionally, each combination was compared to a naïve classifier, one that would classify every sentence as a 0 for each of the events. This would allow us to determine a desired lower bound for the performance of the algorithms.

Display of Results

As mentioned previously, 4000 sentences were used as a test set. Due to the number of combinations of event, feature representation, and algorithm, a large amount of data was produced. For brevity, it was decided that results of classification for Recurrence and Chemotherapy would be investigated in depth. Recurrence was chosen because it was an example of multiclass classification. Chemotherapy was chosen because it was the most pervasive event.

Table 5: Output of various algorithms

5.1: Recurrence

Traditional Bag of Words

Logistic Regression

```
Classification Report
             precision    recall  f1-score   support

         -1       0.38      0.08      0.13        37
          0       0.98      1.00      0.99      3924
          1       0.54      0.18      0.27        39

avg / total       0.97      0.98      0.98      4000
```

Naïve Classifier

```
Classification Report - Naive
             precision    recall  f1-score   support

         -1       0.00      0.00      0.00        37
          0       0.98      1.00      0.99      3924
          1       0.00      0.00      0.00        39

avg / total       0.96      0.98      0.97      4000

Time taken for  Category: recurrence  Method: 1gram      Algorithm: logistic :
4.485043287277222 seconds.
```

Multilayer Perceptron

```
             precision    recall  f1-score   support

         -1       0.10      0.14      0.12        37
          0       0.99      0.99      0.99      3924
          1       0.00      0.00      0.00        39

avg / total       0.97      0.97      0.97      4000

Time taken for  Category: recurrence  Method: 1gram      Algorithm: perceptron :
4.134629249572754 seconds.
```

K-Nearest Neighbors

```
             precision    recall  f1-score   support

         -1       0.75      0.24      0.37        37
          0       0.98      1.00      0.99      3924
          1       0.00      0.00      0.00        39

avg / total       0.97      0.98      0.98      4000

Time taken for  Category: recurrence  Method: 1gram      Algorithm: knn :
3.295135021209717 seconds.
```

Decision Tree

```
             precision    recall  f1-score   support

         -1       0.14      0.19      0.16        37
          0       0.98      0.98      0.98      3924
          1       0.10      0.13      0.11        39

avg / total       0.97      0.96      0.97      4000

Time taken for  Category: recurrence  Method: 1gram      Algorithm: tree :
2.387955904006958 seconds.
```

3-Gram Bag of Words

Logistic Regression

```
             precision    recall  f1-score   support

         -1       0.76      0.59      0.67        37
          0       0.99      1.00      0.99      3924
          1       0.54      0.49      0.51        39

avg / total       0.99      0.99      0.99      4000
```

Naïve Classifier

```
Classification Report - Naive
             precision    recall  f1-score   support

         -1       0.00      0.00      0.00        37
          0       0.98      1.00      0.99      3924
          1       0.00      0.00      0.00        39

avg / total       0.96      0.98      0.97      4000

Time taken for  Category: recurrence  Method: brute_forceAlgorithm: logistic :
37.60845398902893 seconds.
```

Multilayer Perceptron

```
             precision    recall  f1-score   support

         -1       0.73      0.73      0.73        37
          0       0.99      1.00      1.00      3924
          1       0.58      0.54      0.56        39

avg / total       0.99      0.99      0.99      4000

Time taken for  Category: recurrence  Method: brute_forceAlgorithm: perceptron :
171.68322849273682 seconds.
```

TF-IDF Bag of Words

Logistic Regression

```
             precision    recall  f1-score   support

         -1       0.79      0.59      0.68        37
          0       0.99      1.00      1.00      3924
          1       0.67      0.51      0.58        39

avg / total       0.99      0.99      0.99      4000

Time taken for  Category: chemotherapy      Method: tfidf  Algorithm: logistic :
5.400489330291748 seconds.
```

Naïve Classifier

```
             precision    recall  f1-score   support

         -1       0.00      0.00      0.00        37
          0       0.98      1.00      0.99      3924
          1       0.00      0.00      0.00        39

avg / total       0.96      0.98      0.97      4000
```

Multilayer Perceptron

```
             precision    recall  f1-score   support

         -1       0.81      0.68      0.74        37
          0       0.99      1.00      1.00      3924
          1       0.88      0.36      0.51        39

avg / total       0.99      0.99      0.99      4000

Time taken for  Category: recurrence  Method: tfidf      Algorithm: perceptron :
19.318445205688477 seconds.
```

K-Nearest Neighbors

```
             precision    recall  f1-score   support

         -1       0.70      0.19      0.30        37
          0       0.98      1.00      0.99      3924
          1       0.50      0.05      0.09        39

avg / total       0.98      0.98      0.98      4000

Time taken for  Category: recurrence  Method: tfidf      Algorithm: knn :
368.86042952537537 seconds.
```

Decision Tree

```
             precision    recall  f1-score   support

         -1       0.73      0.81      0.77        37
          0       1.00      0.99      1.00      3924
          1       0.71      0.69      0.70        39

avg / total       0.99      0.99      0.99      4000

Time taken for  Category: recurrence  Method: tfidf      Algorithm: tree :
7.4120965003967285 seconds.
```

5.2: Chemotherapy

Traditional Bag of Words

Logistic Regression

```
Classification Report
             precision    recall  f1-score   support

          0       0.95      1.00      0.97      3782
          1       0.57      0.10      0.16       218

avg / total       0.93      0.95      0.93      4000

Time taken for  Category: chemotherapyMethod: 1gram      Algorithm: logistic :
3.2356109619140625 seconds.
```

Naïve Classifier

```
Classification Report - Naive
             precision    recall  f1-score   support

          0       0.95      1.00      0.97      3782
          1       0.00      0.00      0.00       218

avg / total       0.89      0.95      0.92      4000
```

Perceptron

```
             precision    recall  f1-score   support

          0       0.96      0.99      0.97      3782
          1       0.61      0.28      0.39       218

avg / total       0.94      0.95      0.94      4000

Time taken for  Category: chemotherapyMethod: 1gram      Algorithm: perceptron :
3.901289463043213 seconds.
```

K-Nearest Neighbors

```
Classification Report
             precision    recall  f1-score   support

          0       0.95      0.99      0.97      3782
          1       0.60      0.13      0.22       218

avg / total       0.93      0.95      0.93      4000

Time taken for  Category: chemotherapyMethod: 1gram      Algorithm: knn :
3.249997138977051 seconds.
```

Decision Tree

```
             precision    recall  f1-score   support

          0       0.96      0.95      0.96      3782
          1       0.28      0.32      0.30       218

avg / total       0.92      0.92      0.92      4000

Time taken for  Category: chemotherapyMethod: 1gram      Algorithm: tree :
2.398338556289673 seconds.
```

3-Gram Bag of Words

Logistic Regression

```
             precision    recall  f1-score   support

          0       0.98      0.98      0.98      3782
          1       0.72      0.69      0.70       218

avg / total       0.97      0.97      0.97      4000

Time taken for  Category: chemotherapyMethod: brute_forceAlgorithm: logistic :
65.30312395095825 seconds.
```

Naïve Classifier

```
             precision    recall  f1-score   support

          0       0.95      1.00      0.97      3782
          1       0.00      0.00      0.00       218

avg / total       0.89      0.95      0.92      4000
```

Perceptron

```
Classification Report
             precision    recall  f1-score   support

          0       0.98      0.99      0.99      3782
          1       0.79      0.69      0.74       218

avg / total       0.97      0.97      0.97      4000

Time taken for  Category: chemotherapyMethod: brute_forceAlgorithm: perceptron :
171.58691263198853 seconds.
```

K-Nearest Neighbors

```
Classification Report
             precision    recall  f1-score   support

          0       0.97      0.99      0.98      3782
          1       0.78      0.44      0.56       218

avg / total       0.96      0.96      0.96      4000

Time taken for  Category: chemotherapyMethod: brute_forceAlgorithm: knn :
298.4291546344757 seconds.
```

Decision Tree

```
Classification Report
             precision    recall  f1-score   support

          0       0.98      0.98      0.98      3782
          1       0.60      0.61      0.60       218

avg / total       0.96      0.96      0.96      4000

Time taken for  Category: chemotherapyMethod: brute_forceAlgorithm: tree :
42.48824644088745 seconds.
```

TF-IDF Bag of Words

Logistic Regression

```
Classification Report
             precision    recall  f1-score   support

          0       0.98      0.99      0.98      3782
          1       0.76      0.69      0.72       218

avg / total       0.97      0.97      0.97      4000

Time taken for  Category: chemotherapyMethod: tfidf      Algorithm: logistic :
5.400489330291748 seconds.
```

Naïve Classifier

```
             precision    recall  f1-score   support

          0       0.95      1.00      0.97      3782
          1       0.00      0.00      0.00       218

avg / total       0.89      0.95      0.92      4000
```

Perceptron

```
             precision    recall  f1-score   support

          0       0.98      0.99      0.98      3782
          1       0.75      0.71      0.73       218

avg / total       0.97      0.97      0.97      4000

Time taken for  Category: chemotherapyMethod: tfidf      Algorithm: perceptron :
20.071147203445435 seconds.
```

K-Nearest Neighbors

```
                precision    recall  f1-score   support

          0       0.95      1.00      0.97      3782
          1       0.90      0.12      0.22       218

avg / total       0.95      0.95      0.93      4000

Time taken for  Category: chemotherapyMethod: tfidf      Algorithm: knn :
370.56969809532166 seconds.
```

Decision Tree

```
Classification Report
             precision    recall  f1-score   support

          0       0.98      0.98      0.98      3782
          1       0.68      0.67      0.67       218

avg / total       0.96      0.96      0.96      4000

Time taken for  Category: chemotherapyMethod: tfidf      Algorithm: tree :
23.32229518890381 seconds.
```

Chapter 6

Conclusions and Future Work

After evaluating the results above, a few key insights were noted.

Sparsity of the data

Most sentences did not code for any events. That is, most of the information in progress notes is not directly related to these events. Thus, if one can remove these sentences, or perhaps highlight the important ones, physicians would be able to cut down on their time reading charts.

This also made evaluating the algorithms more nuanced. In general, the benchmark naïve algorithm had very high F1-Scores. This was again because of the sparsity of the data. Since around 95% – 98% of sentences did not code for an event, the naïve approach would be able to score incredibly high.

Algorithm Performance

At present, the information above suggests that TF-IDF and 3-Gram BOWs work very well at representing data when fed into logistic regression, multilayer Perceptrons, K-Neighbors, and decision trees. However, Logistic Regression and Multilayer Perceptrons seemed to outperform the other methods.

This is rather puzzling, as logistic regression is considered a simpler and less powerful machine learning method, especially compared to the Multilayer Perceptron. However, for this set of data (sparse bags of words) it seems that logistic regression can hold its own. Further investigation would be required to understand why it works so well in this particular domain.

6.1: Additional Work

Improving data creation

The most time-consuming part of the project was manually coding the sentences to produce the test data. In order to bring make this pipeline scalable, it is important to address this issue. One promising solution is the implementation of distance learning. Specifically, the creation of more positive sentences (those that code events as 1s rather than 0s) would be beneficial, in order to deal with the sparsity problem present in the current dataset.

Using other Feature Representations

Although other feature representations were discussed, only the BOW, TF-IDF BOW and N-Gram BOWs were used in training and testing the learning algorithms. It would be beneficial to apply the other feature representations to the algorithms, either by themselves or in conjunction with the BOWs to see if any improvements could be made.

Collaboration

If anything, the most important part of this project was the creation of the coded dataset. Because it was time consuming process, being able to share that dataset would be beneficial to anyone pursuing event extraction in a clinical setting.

At this point, we are in contact with a research group based at the University of Victoria who are exploring a similar problem. However, they are using the output of a prebuilt NLP system called cTakes built by the Apache Software Foundation. This program uses SNOMED categorizations of clinical nomenclature, and recognizes certain instances of text. By supplying the dataset to them, we hope that more people can benefit from the work done in this project.

6.2: Future applications

There are some key applications that can be pursed with the results of this project. By being able to extract event information from unstructured text, information systems can be developed to streamline physician workflows by refining the information they need to consume prior to seeing a patient.

One such tool is a clinical decision support system (CDSS), which augments a physician's ability to analyze and understand patients. These are "*any software designed to directly aid in clinical decision making in which characteristics of individual patients are matched to a computerized knowledge base for the purpose of generating patient specific assessments or recommendations that are then presented to clinicians for consideration*." (Hoyt & Yoshihashi, 2014). In order for such a tool to be feasible, a majority of patient information would need to be uncovered from unstructured data. However, as with this project, scalability would be a key limiting factor.

Bibliography

Lawrence TS, Ten Haken RK, Giaccia A. Principles of Radiation Oncology. In: DeVita VT Jr., Lawrence TS, Rosenberg SA, editors. Cancer: Principles and Practice of Oncology. 8th ed. Philadelphia: Lippincott Williams and Wilkins, 2008

Contextual Bag-of-Words for Visual Categorization
(Li, Tao, & Xian-Sheng, 2011)

Bag-of-words representation for biomedical time series classification
(She, Nahavandia, Kouzani, Wang, & Liu, 2013)

A Survey of event extraction methods from text for decision
support systems
(Kaymak, de Jong, Caron, Hogenboom, & Frasincar, 2016)

Health Informatics - Practical Guide for Healthcare and Information Technology Professionals
(Hoyt & Yoshihashi, 2014)

An Introduction to Statistical Learning with Applications in R
(James, Witten, Hastie, & Tibshirani, 2014)

Natural Language Processing with Python
(Bird, Klein, & Loper, 2014)

Microsoft Research – Relation Extraction
https://www.microsoft.com/developerblog/real-life-code/2016/09/14/Relation-Extraction-Python.html

SPACY
https://spacy.io/

NLTK
http://www.nltk.org/

SciKit-Learn
http://scikit-learn.org/stable/index.html

cTAKES
https://ctakes.apache.org/