The University of Manchester

Extracting Social Factors from Clinical Narratives

A dissertation submitted to The University of Manchester for the degree of
Master of Science
in the Faculty of Science and Engineering

Andika Yudha Utomo

School of Computer Science

2016

List of Contents

Word count : 20,117

List of Tables

List of Figures

List of Equations

List of Abbreviations

Abbreviation	Definition
ANNIE	A Nearly-New Information Extraction System
CAD	Coronary Artery Disease
CRIS	Clinical Record Interactive Search
CSV	Comma-Separated Values
DFD	Data Flow Diagram
GATE	General Architecture for Text Engineering
GNU	GNU is Not Unix
GUI	Graphical User Interface
i2b2	Informatics for Integrating Biology and the Bedside
JAPE	Java Annotation Pattern Engine
k-NN	k-Nearest Neighbours
LaSIE	Large Scale Information Extraction
LHS	Left Hand Side
MoSCoW	Must have, Should have, Could have, and Would like but won't get
NHS	National Health Service
ppd	packs per day
Regex	Regular Expression
RHS	Right Hand Side
SVM	Support Vector Machine
UK	United Kingdom
XML	Extensible Markup Language

Abstract

The data deluge phenomenon brought its impact to medical practice. In recent years, health institutions have started to transform their patients' clinical narratives from the handwritten to the digital format. This condition opens the possibility to develop automated systems to analyse clinical records further.

One of the possibilities to be examined from clinical narratives is patient's social factors, such as smoking and alcohol consumption. This project aimed to develop a rule-based text mining system to extract smoking status from clinical narratives. The system was also repurposed to extract alcohol consumption status.

Two datasets were obtained from the Informatics for Integrating Biology & the Bedside (i2b2) repository for the development purpose. A text mining system was designed and developed in four iterations by using the General Architecture for Text Engineering (GATE) platform and its plugins: Java Annotation Pattern Engine (JAPE) and A Nearly New Information Extraction System (ANNIE). Some additional scripts have also been developed by using Java programming language to support the workflow.

The system was evaluated with two test datasets from the i2b2 repository. In addition, a dataset from The Christie NHS Foundation Trust in Manchester was retrieved to test out the reusability of the system to handle data from different sources. The evaluation resulted microaveraged F-measure scores of more than 0.90 on average for both smoking and alcohol consumption extractions, which can be considered as a state-of-the-art performance for a rule-based system.

It can be concluded that the system generalised well on the data from different sources. Furthermore, the system can be repurposed to handle other social factors such as drug abuse, medication, family, and pet histories. The future work will be integrating these social factors to predict health related quality of patient's life.

In addition, the system was published as an open source program on the GitLab repository of the University's School of Computer Science, and is in a process to be published on the GitHub repository of The Christie NHS Foundation Trust.

Keywords: *clinical text mining, rule-based system, social factors extraction.*

Declaration

No portion of the work referred to in the dissertation has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Intellectual Property Statement

i. The author of this dissertation (including any appendices and/or schedules to this dissertation) owns certain copyright or related rights in it (the "Copyright") and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this dissertation, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has entered into. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the "Intellectual Property") and any reproductions of copyright works in the dissertation, for example graphs and tables ("Reproductions"), which may be described in this dissertation, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this dissertation, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/display.aspx?DocID=24420), in any relevant Dissertation restriction declarations deposited in the University Library, and The University Library's regulations (see http://www.library.manchester.ac.uk/about/regulations/_files/Library-regulations.pdf).

Acknowledgements

"Standing on the shoulders of giants"

I recalled this classic metaphor when I was reaching the conclusion part of this dissertation report. It means that to build a more advanced knowledge, we should rely on the previously discovered knowledge. I realised that my work was just a microscopic part of the advancing field of clinical text mining. My hopes are simple: someone will read this dissertation, improve it, and make the "future work" section a reality. I hope that this research will be beneficial to others.

***

All praise belongs to God, The Most Gracious, The Most Merciful, and The All Knowing. Only with the help from Him I can finish this master's programme successfully. I prayed that the knowledge we have obtained thus far will ease our ways toward Him.

In this occasion, I would like to extend my gratitude to those who supported me to accomplish my master's study:

1. The Indonesia Endowment Fund for Education (in Indonesian: Lembaga Pengelola Dana Pendidikan – LPDP), which supported my one-year study in the UK. I owe a lot of contribution to make a better Indonesia upon finishing my study.
2. Dr. Goran Nenadic and Dr. Azad Dehghan, who introduced me to this field and have been supporting this research from the very beginning until the end. I received a lot of insights and suggestions from them.
3. The Christie NHS Foundation Trust in Manchester and its Data Science Team, for allowing me to use their anonymised clinical records for this research.
4. Azzahra Ulya, my life partner to be, who has been very patiently waiting for me this one year. May God ease our goodwill to build a civilised society, together.

5. Yodha, Hapsoro, Rachma, Ryan, and Danur – my Indonesian partners in the School of Computer Science, who have been with me, passing through ease and hardship together this one year.
6. Zulfikar, Bening, Beni, and Rian – my traveling mates who have thousands of motivational stories to share. You all enriched my experience in the UK.
7. Indonesian Student Society in Greater Manchester, which reminds me constantly about Indonesia, helps me to cope with this homesickness.
8. All of other parties which I cannot mention one by one.

I believe that there is no coincidence in every single moment we had in our life, including our meetings. I wish the best success for you all in your future endeavours.

Manchester, September 4th, 2016

The Author

I extend my gratefulness and appreciation to my mother. She might not understand what this dissertation report is, but her support for me to follow my vision has been a very meaningful spirit for me.

1. Introduction

Many human activities involve data generation, not the least of which is medical practice. An example is the use of electronic health records by health practitioners that include unstructured data sources to note specific clinical events in a patient's healthcare process [1]. A single health institution can generate a large amount of medical data. As an instance, the Clinical Record Interactive Search (CRIS), that contains data from the South London and Maudsley NHS Foundation Trust, has over 20 million clinical narrative records and continues to grow every month [2]. The data is predicted to grow at an exponential rate as healthcare systems are developed [3]. To cope with this data deluge, an automated process is needed to derive information and knowledge from such kind of data. This field is known as clinical text mining [2].

One of the challenges in clinical text mining is extracting social factors, such as smoking and alcohol consumption status, from clinical narratives. The extraction of social factors can be used as contextual information by a physician to analyse specific diseases. The automation of this process will shorten the time needed for physicians to obtain the information about social factors of their patients, as this process is normally conducted by a nurse who reads the patient's clinical notes manually. The automation also enhances patients' convenience as the physician needs less time to give them feedback of their medications [4]. In addition, the automation will ease the task of data processing and retrieval on a large scale, which can be used to advance the research in medical practice.

Informatics for Integrating Biology and the Bedside (i2b2) [5] addressed this task in 2006 by organising a challenge to identify smoking status from clinical narratives. The participants were given a set of clinical records for development. They were asked to develop a system in a given time frame to be evaluated with testing data that is unseen until the end of the competition [6]. This dissertation project utilises the challenge as a starting point to develop a social factor extractor from clinical narrative. The previous attempts will be examined and enhanced to develop a state-of-the-art social factor extraction system that works for various types of clinical narratives.

1.1. Aim

The aim of this research is to develop a text mining system that takes an input of unstructured clinical records of patients and generates an output of the extracted smoking and alcohol consumption statuses. Rule-based approaches will be used in this project. Datasets from the i2b2 repository will be obtained for development and testing purposes. In addition, a dataset from The Christie NHS Foundation Trust in Manchester will be used to test the system in a real-life situation.

1.2. Objectives

1.2.1. Learning Objectives

1) Study the concepts and trends of text mining and information extraction along with the particular issues of clinical text mining.
2) Explore the i2b2 2006 challenge of smoking status identification and its successful attempts.
3) Learn the rule-based approaches of information extraction.
4) Explore the General Architecture for Text Engineering (GATE) platform, along with the Java Annotation Pattern Engine (JAPE) language that will be used to develop a rule-based information extraction system.

1.2.2. Deliverable Objectives

1) Obtain the required train datasets from the i2b2 data repository.
2) Investigate the characteristics of how social narratives are expressed within clinical records.
3) Design a text mining workflow to extract social factors by using rule-based approaches.
4) Develop a system based on the workflow by utilising the GATE environment.
5) Develop extraction rules by using JAPE based on the train datasets.
6) Test out the system with the test datasets from the i2b2 repository and a dataset from The Christie Hospital in Manchester.
7) Report the performance of the system.
8) Publish the system as an open source software.

9) Report the recommendations for future research.

1.3. Report Structure

This dissertation report consists of seven chapters:

Chapter 1 – Provides the introduction, aim, and objectives of this project.

Chapter 2 – Discusses the literature study of text mining and the previous attempts to solve similar problems.

Chapter 3 – Includes the limitations, requirements of the system, and a deep discussion about the methodology of this research.

Chapter 4 – Explains the system architecture based on the foundation in the previous chapters.

Chapter 5 – Provides a detailed discussion about system development stages.

Chapter 6 – Discusses the system evaluation with the test datasets from the i2b2 repository and an additional dataset from The Christie NHS Foundation Trust in Manchester.

Chapter 7 – Concludes the research and discusses some insights for future development.

2. Background

This section discusses the background research regarding text mining and its application in clinical practices as the primary topic of this project. Furthermore, the previous attempts to extract social factors from clinical narratives will be discussed and analysed to give an insight into how this project will be done.

2.1. Text Mining

As the advancement of computer technology spreads through broad aspects of society, the amount of data generated also has increased. This condition is understandable since data is often required in the computing process. Furthermore, research shows that more than 80% of today's data are in the unstructured format [7]. Unfortunately, the usual logic-based computing paradigm cannot handle this type of data properly, as free text often contains confusing and ambiguous meanings [8]. This condition leads the development of a data mining branch that focuses on unstructured text, known as text mining. It aims to discovering information in free-form text documents by annotating them for the presence of certain contents or relationships [9].

The term text mining began appearing in about 1999. Almost no practitioners were in this field at that time, but people began to recognise the potential importance of the field. Since then, innovations in text mining have rapidly expanded to cope with the need for extracting critical information in unstructured text. Researchers started to combine text mining with other analytical sciences such as data mining, statistical analysis, and other related fields to excavate potentially valuable information contained in documents [9].

Generally, a text mining system takes the input of an unstructured document and generates a wide variety of output depends on the development purpose. The output can be a classification result, graphs, maps, or any other forms [10]. A generic approach to text mining is divided into two parts: text pre-processing phase for initial harmonisation of text and information extraction phase to obtain important information from that text.

2.1.1. Text Pre-processing

Before the text is processed further, it must be segmented into simpler forms. The process of splitting text into smaller parts typically occurs at the level of paragraphs, sentences, words, or even syllables depending on the purpose of a system. As for text mining, the most commonly used approaches are separating the text into sentences (sentence splitting) and basic units (tokenisation) [10]. This step is done to ease further processes by providing a simplified form of text that can be easily managed by computer.

Sentence splitting and tokenisation are not trivial tasks since every language has its rules. As for the English, one of the challenges is to determine the proper intention of punctuation marks. As an instance, a sentence splitter should distinguish whether or not a period indicates the end of a sentence [10].

Table 2.1 shows some examples of sentences taken from clinical narratives that contain ambiguous period usage such as to mark numbers, alphanumeric references, and abbreviations [5] [11]. While humans can easily recognise whether a period indicates the end of a sentence, it is not a straightforward task for a machine. The programmer should cover a number of cases where a period is meant to end a sentence, as well as where it is not, when designing a sentence splitter module.

Table 2.1. Examples of ambiguous periods in a sentence

Sentence
Follow-up will be with Dr. Shock , her primary hematologist-oncologist , on 04/13/98 .
The electrocardiogram revealed atrial fibrillation at 128 per minute with no detectable P-R interval , QRS of 0.09 , and QTC of 0.42 .
8. Compazine 10.00mg p.o. q 6 h. p.r.n. for nausea .

Another issue of text pre-processing is the tokenisation process. A token is defined as the smallest component in text, which could be a word, punctuation, or abbreviation. Sometimes the process is not straightforward because of the variation of the language itself.

Some examples of tokenisation are shown in Table 2.2. In the first sentence, the word "*doesn't*" is problematic as it is a contraction of two words "*does not*". The developer should consider whether to treat it as a single token "*doesn't*" or to divide it into two tokens as "*does*" and "*n't*". Another problem in the second sentence is how to handle date mentions. The first representation groups the whole date as a single token, while the latter divides it into numbers and slashes. While all of the representations are acceptable, the developers should be aware of how their system works, as the details may affect further processes. If it is possible, the developer can also choose which tokenisation style provides the most advantages to the system that will be developed.

Table 2.2. Examples of tokenisation variation

Sentence	Tokenisation
He doesn't smoke.	*1) He doesn't smoke .*
	2) He does n't smoke .
The patient should come back on 12/12/2016.	*1) The patient should come back on 12/12/2016 .*
	2) The patient should come back on 12 / 12 / 2016 .

The issue of the usage of punctuation marks in a text has been raised because texts in different knowledge domains could have different styles, which complicates the problem. This condition encouraged text mining researchers to propose their own methods to tokenise text. Some of them are Dridan & Oepen (2012), who proposed a rule-based tokenisation that allows a flexible configuration [12], and Marsik & Bojar (2012), who developed a classifier-based tokenisation enhanced with an ability to consider prerequisites configured by users [13]. Developers then select a specific tokenisation method that fits their text style and development purpose.

2.1.2. Information Extraction

After the text has passed into the pre-processing stage, the next step is to obtain crucial information as defined in the requirements. This process of manipulating the text to obtain valuable information is known as information extraction. Some

approaches are available for extracting information from text; they include rule-based approaches, machine learning-based approaches, and a hybrid method that combines both of the mentioned approaches. This section presents an overview of the rule-based approaches, as this type of approach is the main method used in this project.

The rule-based approaches require experts in a specific knowledge domain to manually develop rules based on requirements and data [14]. Some techniques can be used to develop rules: heuristic approaches (if-else statements), regular expressions, or a more advanced rule-based language such as Java Annotation Pattern Engine (JAPE) that is embedded in the General Architecture of Text Engineering (GATE) platform [2] [15]. Rule-based approaches can deliver an accurate extraction from the text because the rules are carefully crafted based on the task given. However, it is a labour-intensive and expensive work, as the process is done manually by experts. If the requirements change, the experts must adjust the rules to the new requirements. Below is an example of the rule-based approach developed using JAPE [16].

```
Rule: TemporalSentences

(

      {Token.string =~ "minute|hour|day"}

): predictor

-->

:predictor.TemporalWords = {expression = temporal}
```

Figure 2.1. Example of a simple JAPE rule

The example in Figure 2.1 shows a simple JAPE rule to identify particular keywords in the text. It tries to identify temporal tokens that contain a substring *minute*, *hour*, or *day* and classify them into `TemporalWords` with a feature named `expression` that contains `temporal` as the value. A JAPE rule can be very simple or very complex, depending on the purpose and the design. Typically, an information extraction system consists of several JAPE rules that are executed consecutively to support each other. This modularity concept is better than designing a very long rule to catch all the expressions in one turn. The modularity will enhance the code readability, thus make the debugging process more convenient if errors or changes in the requirements occur.

2.2. Clinical Text Mining

Clinical text mining is one of the branches of text mining that is growing rapidly in recent years. Numerous studies have attempted to explore the possibilities of text mining to enhance the effectiveness and efficiency in various health-related aspects, including symptom identification, drug discovery, gene relation extraction, temporal relation mining, and social factor extraction. The advancement of this field was also triggered by the public challenges held by various research institutions. Huang and Lu (2016) summarised the public challenges of clinical text mining from 2002 to 2014 [17].

2.2.1. Public Challenges

Several public challenges have been presented annually to solve particular problems arising in medical practice. Some of them are BioCreative [18], which has provided various challenges in medical biology since 2004, BioNLP-Shared Task [19], which provides annual tasks of information extraction of various subjects, and Informatics for Integrating Biology and the Bedside (i2b2) [5], which provides various tasks related to de-identification and information extraction of clinical narratives. These challenges are typically simplified forms of real-world conditions. This simplification is done to target specific details and to ease the evaluation process of submitted attempts [17].

One of the challenges published by i2b2 in 2009 was medication information extraction from clinical text [20]. This challenge aimed to encourage the development of systems to extract medication-related components provided patients' medical records. These components include medications, dosages, and frequency of administrations. Twenty teams from nine countries took part in this challenge, with an overall F-measure score of ten best submissions ranging from 0.76 to 0.86 [21].

Another example of challenge by i2b2 is the identification of risk factors for heart disease over time [22]. Given the anonymised medical records of diabetic patients that contain information about possible heart disease risk factors, participants were asked to identify relevant heart disease risk and track patients' progression over a set of their clinical records over time. There were 29 attempts submitted to the challenge; six of them scored over 0.90 for F-measure [23].

Aside from these two challenges, many other competitions aim to develop text mining systems to solve specific tasks. Generally, the task providers give a certain amount of freedom for the participants to develop the system. This flexible condition encourages the participants to creatively formulate methods and algorithms to solve the problems. These kinds of challenges also boost collaboration of the researchers to share their thoughts and ideas about the methodology to solve certain types of problems [17]. Additionally, anonymised datasets are often released as a part of the challenges. The data, which typically require a long process to obtain due to privacy concerns, can be used for educational or research purposes. As a result, a number of novel methods to solve various problems in clinical text mining were discovered. These facts will advance the future development of this field.

2.2.2. Techniques Used in the Challenges

The submissions to the challenges can be classified into three main categories of text mining approaches: rule-based approaches, machine learning-based approaches, and hybrid approaches. The top performing systems generally are those built using hybrid approaches, followed by rule-based approaches and machine-learning approaches [21] [23]. In addition, the type of approach used is not the only factor that affects the quality. For example, the submissions from the University of Wisconsin-Milwaukee and the University of Sydney for the medication information extraction challenge were using the Support Vector Machines (SVM) in their workflows, but their overall F-measures were significantly different. The first one gained an F-measure score of 0.857, while the other gained 0.764 [21]. This condition also occurred for the rule-based and hybrid approaches. To conclude, the same methods do not necessarily produce a similar result. There are other factors that hold important roles in determining the quality of developed systems, including the engineering details, features used, and workflow designs [6].

2.2.3. Outcomes

Different clinical text mining challenges have different state-of-the-art performance results. For example, the top performers' F-feasure scores of the i2b2 challenge in medication information extraction was approximately 0.80, while the score reached over 0.90 in the heart disease risk factors identifications challenge [21] [23]. These

results proved that various tasks in clinical text mining have different levels of difficulties and challenges.

Various public challenges in clinical text mining indicate the importance of this field for application in the real world. These challenges also led to the development of more advanced algorithms or tools to undertake specific tasks [17]. Many submissions benefited from the previous attempts in similar challenges; thus, the developed systems are generally more robust compared to the previous systems.

2.3. Social Factors Extraction from Clinical Narratives

This dissertation research is based on a public clinical text mining challenge provided by the i2b2 in 2006. The challenge was to identify smoking status from unstructured clinical records [6]. The smoking condition in a clinical record should be grouped into one of five categories: CURRENT SMOKER, PAST SMOKER, NON-SMOKER, SMOKER, and UNKNOWN. Three of the successful attempts are discussed in this part, including the attempts made by Aramaki *et al.* [24], Cohen [25], and Clark *et al.* [26]. Those attempts were developed using hybrid methods, a combination of rule- and machine learning-based approaches.

2.3.1. Pre-processing Stage

Given the unstructured clinical narratives of a number of patients, the participants were asked to extract the smoking statuses from the narratives. One record of clinical narrative roughly consists of 600-800 words that describe a patient's conditions, diagnose, medication history, and any other relevant medication information. In fact, the smoking status was typically written in just one or two sentences. The participants recognised this condition and took advantage by examining the possible passages that contained smoking status and extracting them from whole records to be processed further. Aramaki *et al.* [24] accomplished this by extracting sentences that contain possible keywords (e.g. nicotine, smoker, smoke, smoking, tobacco, cigarette). If these keywords were not found in the record, it was simply categorised as UNKNOWN and excluded from the further process. Cohen [25] and Clark *et al.* [26] performed a similar process. In addition, to increased performance, Clark *et al.* extended the dataset using their own data. The i2b2 dataset initially consisted of 398

documents before it was extended by 4,294 additional medical reports. This addition was done due to the fact that an error rate of a learning system generally can be reduced by increasing the sample size [27].

2.3.2. Rule-based Approaches

The participants utilised rule-based approaches in similar ways. Aramaki *et al.* [24], Cohen [25], and Clark *et al.* [26] designed heuristic rules to filter out the documents without specific mentions about smoking and categorised them as UNKOWN without passing them through further steps. In addition, Clark *et al.* used heuristic rules to prioritise the smoking mentions. For example, UNKNOWN is less specific than SMOKER and NON SMOKER, and SMOKER is less specific than CURRENT SMOKER and PAST SMOKER. These rules were used to determine the smoking status of a patient based on keywords and temporal expressions found in the record. As for Cohen, he developed some heuristic rules for a final prediction after the data passed into a machine learning algorithm. The rules checked the possible inconsistencies in the classification. For example, if a document is categorised as SMOKER, but "*no history of smoking*" is found in that document, then the prediction will be changed to NON SMOKER.

Figure 2.2 shows an example of rule made by Aramaki *et al.* [24].

```
if a record contains smoking related keywords
        classify as NON-SMOKER
else classify as UNKNOWN
```

Figure 2.2. Example of a baseline rule made by Aramaki et al. [24]

The pseudo-code in Figure 2.2 means that if a smoking related keywords (e.g. smoke, tobacco, cigar) are found in a record, classify it as NON-SMOKER. Otherwise, classify it as UNKNOWN. This is the initial rule made by Aramaki *et al.* before the record processed further to re-classify the NON-SMOKER to a more suitable category. NON-SMOKER was chosen because it is the most probable category if a smoking related keywords are found in record [24].

2.3.3. Machine Learning Approaches

Two machine learning approaches will be discussed in this section. The first one is k-Nearest Neighbour (k-NN) that was used by Aramaki *et al.* [24]. The k-NN algorithm is considered to be one of the most straightforward learning algorithms. This algorithm calculates the distance between a data point *n* and a number of *k* of its nearest neighbours, and then consider *n* to have the same class as its most common neighbours [28]. Aramaki *et al.* extracted the sentences that contain smoking status from the training and testing set, and then calculated the similarity (or in this case, distance) between the classified sentences in the training set and the unclassified testing set. They had experimented with various *k* numbers before establishing $k = 10$ as the best performing algorithm.

Another popular machine learning approach used in this challenge was the Support Vector Machines (SVMs) algorithm, which was used by Cohen [25] and Clark *et al.* [26]. This algorithm works by constructing hyperplanes that produce the largest distance between data points in different classes. Margins are used along the hyperplanes to make sure that it produces the largest distance between classes. Data points that lie in the margin are called support vectors [29]. Cohen realised that SVM has a problem when classifying imbalanced datasets that have one class that is more common than the others. SVM will favour the more common class in the classification process. To overcome this problem, the weight parameter was used. This parameter is calculated for each class to measure its relative rarity, and thus affects the decision made by the algorithm.

One of the factors that determines the quality of a machine learning algorithm is feature selection. Clark *et al.* [26] proposed various features for their SVM algorithm, such as smoking related keywords, temporal expressions, section headings, and linguistic elements. A subset of features was selected by using a feature selection method to determine the best possible combination from existing features.

2.3.4. i2b2 2006 Challenge Results

The three submissions discussed in the previous section were among the best in the challenge. Various approaches were used with solid justifications. The results summary of these submissions can be seen in Table 2.3. The detailed information

about the evaluation parameters (i.e. precision, recall, and F-measure) can be seen in section 3.3.5.

Table 2.3. The 2006 i2b2 challenge top submissions evaluation

Rank	Group	Macroaveraged			Microaveraged		
		Precision	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
1	Clark	0.81	0.73	0.76	0.90	0.90	0.90
2	Cohen	0.64	0.67	0.65	0.88	0.89	0.89
3	Aramaki	0.64	0.67	0.65	0.88	0.89	0.89

The challenge provider used two types of measurements to evaluate the performance of the submissions. Macroaveraged scores give an equal weight to each smoking category, while microaveraged scores give an equal weight to each document. As the datasets given were imbalanced, the scores between these two calculations differed significantly. The dataset contains documents with UNKNOWN status for more than 60%. Most of the submissions recognised this category easily by the absence of smoking-related keywords. On the other hand, the remaining four categories were ambiguous, particularly the SMOKER status. The training data contains only nine records (out of 398) which were classified as SMOKER. This is the most ambiguous category in the dataset since the evidence is insufficient to develop rules or to train machine learning algorithms to recognise the pattern of this smoking category. The distribution of the dataset can be seen in Table 2.4.

This dataset represents the real-world situation in which the distribution along the labels is not necessarily balanced. This is an additional challenge for the participants in figuring out the mechanisms needed to handle such datasets. Aramaki *et al.* [24] recognised the weakness of their system to handle rare words in the dataset. The training set contains only 2% of the records which were classified as SMOKER. To handle this condition, domain experts can be employed to predict possible keywords that lead to SMOKER or any other categories that are not covered appropriately in the dataset. Another solution is to increase the number of training data, which was done

by Clark *et al*. [26]. These two solutions have their own advantages and drawbacks in terms of effort, resource, cost, and time needed to develop the system.

Table 2.4. Training and testing data statistics of the 2006 i2b2 challenge

Smoking Status	Training Data (%)	Testing Data (%)
CURRENT SMOKER	35 (9%)	11 (11%)
PAST SMOKER	36 (9%)	11 (11%)
NON-SMOKER	66 (17%)	16 (15%)
SMOKER	9 (2%)	3 (3%)
UNKNOWN	252 (63%)	63 (61%)
Total	*398 (100%)*	*104 (100%)*

Another aspect that leads the participants to high scores is the recognition of the characteristic of the dataset. By understanding that the smoking status only appears in one or two sentences, the participants extracted these sentences and ignored the others. This method of selecting sentences is crucial to improving the performance of the system.

This dissertation project is based on the i2b2 2006 challenge. Previous attempts have been examined and analysed to develop a more advanced system in recognising social factors mention (i.e. smoking and alcohol consumption status) in clinical narratives. Rule-based approaches will be used in this research for two reasons. First, the previous attempts proved that rule-based systems can provide high performance for this task. Second, insufficient amount of annotated datasets exists to support a machine learning-based system.

In addition, the training data will be extended with the i2b2 2014 challenge dataset to increase the generality of the system. The system will also be tested with an unseen dataset from a different source: The Christie NHS Foundation Trust in Manchester, to evaluate the performance of the system in a real-word situation.

3. Research Methodology

This chapter discusses the research methodology that was used in this project. First, the project scope will be explained, followed by the project requirements. Subsequently, a detailed description of the data is presented. These descriptions are the fundamentals of the text mining phases that will be discussed at the end of this chapter.

3.1. Project Scope and Limitation

This project aims to make a software system that extracts smoking status from unstructured clinical narrative documents with a specific format. At the end of the development, the algorithm will be extended to extract alcohol consumption status given documents with the same format, assuming that the alcohol consumption characteristics are expressed in similar ways to the smoking status. The project only concerns the clinical documents written in English. The extraction will be limited to explicitly mentioned smoking and alcohol consumption characteristics in each document. For the smoking status extraction task, this project is only concerned with tobacco smoking. Other types of smoking, such as marijuana or electric smoking, are not considered. The classification will be performed in record-level, which means that each clinical narrative record of a patient will have one prediction for smoking and alcohol consumption status.

The project focuses on algorithm development to perform such tasks. It is neither intended to make a Graphical User Interface (GUI) of the system nor to consider the data security aspects, as the whole project will be done in offline mode. Users should keep the confidentiality of the data by themselves.

The project is also not concerned in handling various data formats. The data supplied to the system should match, or be modified to, a specific format. A discussion about format conversion will be included, as the project uses data from different sources that need to be transformed to match the system requirement.

3.2. Project Requirements

This section discusses the specification of the system to be developed, along with the characteristics of data that will be used in the project.

3.2.1. System Specification

The main system specification in this project was adopted from the i2b2 2006 challenge of smoking status extraction from clinical narratives, with some adjustment based on the project scope, time availability, and difficulty level. Given a set of clinical narrative data, the system should be able to extract smoking and alcohol consumption status for each record in the data, as shown in Figure 3.1.

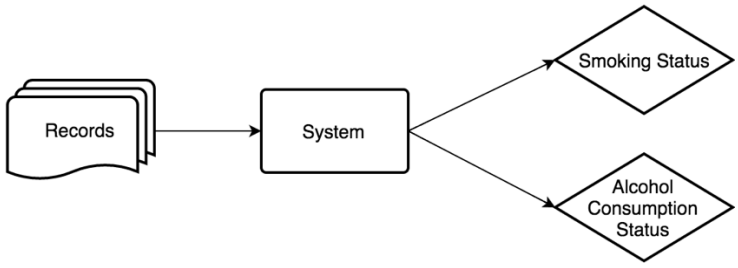

Figure 3.1. The diagram shows that the developed system should be able to extract smoking and alcohol consumption status from clinical narratives

The input document is a text in a specific XML format which contains a number of clinical narrative records. The system is purposed to extract smoking and alcohol consumption status of each record based on explicitly mentioned characteristics.

After the main goal of this project is defined, the MoSCoW schema [30] is then used to breakdown the requirements and to design the level of priority of each requirement. Table 3.1 explains the components in the MoSCoW schema.

Table 3.1. The components of MoSCoW schema [30]

Priority	Explanation
Must Have	The system must able to deliver this function as it is the main or critical functions to make the system work as expected
Should Have	The function is important but not vital
Could Have	The function is wanted but less important

Won't Have (for This Time)	The function is desirable but not important at this point

The MoSCoW schema prioritises tasks based on their importance and desirability. The requirements of this project and their complexities, which are shown in Table 3.2, were developed based on the MoSCoW schema components.

Table 3.2. The requirements of this project

No.	Requirement	Priority	Complexity
1	The system is able to convert the dataset into a suitable format	Should have	Medium
2	The system is able to do appropriate pre-processing mechanisms to recognise tokens and sentences that contain possible smoking (and alcohol consumption) status	Must have	Medium
3	The system is able to take the pre-processed records and determine appropriate *smoking status*	Must have	High
4	The system is able to take the pre-processed records and determine appropriate *alcohol consumption status*	Should have	High
5	There is a function to evaluate the performance of the developed system in terms of accuracy and its related measurements (precision, recall, F-measure)	Must have	Medium
6	The system can be executed conveniently through the existing framework (GATE)	Could have	Medium
7	There is a dedicated graphical user interface that users can run the system	Won't have	High
8	The system is secure, the data is transmitted in an encrypted mode throughout the process	Won't have	High

3.2.2. Data Specifications

Two train datasets are used in this project and will be obtained from the i2b2 data repository. The first one is the corpus of the i2b2 2006 challenge of smoking status identification. It contains 398 records of annotated training data. The latter is from the i2b2 2014 challenge of heart disease risk factors over time identification. The corpus contains 790 records of annotated training data, almost double the size of the i2b2 2006 data. However, 19 records must be excluded, as they do not have smoking status annotated within them. Each record contains about 500 to 800 words that explain the patient's condition at the time the record was written.

The i2b2 2006 challenge defined five types of smoking status in the data based on the condition of patients (smoking or not) and the time they have smoked (past, current, ever). This status is attached to each record in the dataset. The description of each status can be seen in Table 3.3.

Table 3.3. The five smoking statuses and an example of each status

Smoking status	**Characteristic**
	Example
CURRENT SMOKER	The patient that was a smoker within one year
	She is a heavy smoker, not stopping until now.
PAST SMOKER	The patient was a smoker one year or more ago, has not smoked for at least one year
	She is a past smoker, quit two years ago.
NON-SMOKER	The patient who never smoked
	No tobacco.
SMOKER	The patient who is either PAST SMOKER or CURRENT SMOKER, but the information is insufficient to classify the patient into one of the two categories
	History of tobacco use
UNKNOWN	There is no mention of smoking status in the document
	-

The smoking status labels used in the i2b2 2014 challenge were slightly different. Nevertheless, according to the challenge paper [31], the description of each status of i2b2 2014 is comparable to the status in i2b2 2006. For simplicity purposes, the smoking status naming of i2b2 2014 will be converted to match that of i2b2 2006. The conversion is presented in Table 3.4.

Table 3.4. Smoking status naming conversion for i2b2 2014 data

i2b2 2014 naming	**i2b2 2006 equivalent (used in this project)**
current	CURRENT SMOKER
past	PAST SMOKER
never	NON-SMOKER
ever	SMOKER
unknown	UNKNOWN

Both datasets contain unbalanced record numbers for each smoking status. Most of the records are in the UNKNOWN status, where there is no discussion of smoking condition is mentioned. The next dominant status is NON-SMOKER, followed by PAST SMOKER and CURRENT SMOKER. SMOKER (with no indication whether it is PAST or CURRENT) is the status with the least frequency, as this status was only given if information is insufficient to mark the record as PAST SMOKER or CURRENT SMOKER, or an ambiguity is present in the record that made the annotators unsure about the current condition of the patient. The data distribution of both datasets can be seen in Table 3.5.

Table 3.5. Frequency of the record of each smoking status in the training data

Smoking Status	**i2b2 2006 (%)**	**i2b2 2014 (%)**
CURRENT SMOKER	35 (8.8%)	58 (7.5%)
PAST SMOKER	36 (9.0%)	149 (19.3%)
NON-SMOKER	66 (16.6%)	184 (23.9%)

SMOKER	9 (2.3%)	9 (1.2%)
UNKNOWN	252 (63.3%)	371 (48.1%)
Total	*398 (100%)*	*771 (100%)*

Some differences can be seen in the distribution of the i2b2 2006 and 2014 data. The i2b2 2014 data is more likely to have a smoking status note of the patient since the records were taken from the patients with a risk of Coronary Artery Disease (CAD), the most common type of heart disease. It is also more likely that the patients were PAST SMOKERS, since they were already aware about the condition of their health [23]. This situation can be seen in Figure 3.2 which compares the relative distribution of each data label of both datasets.

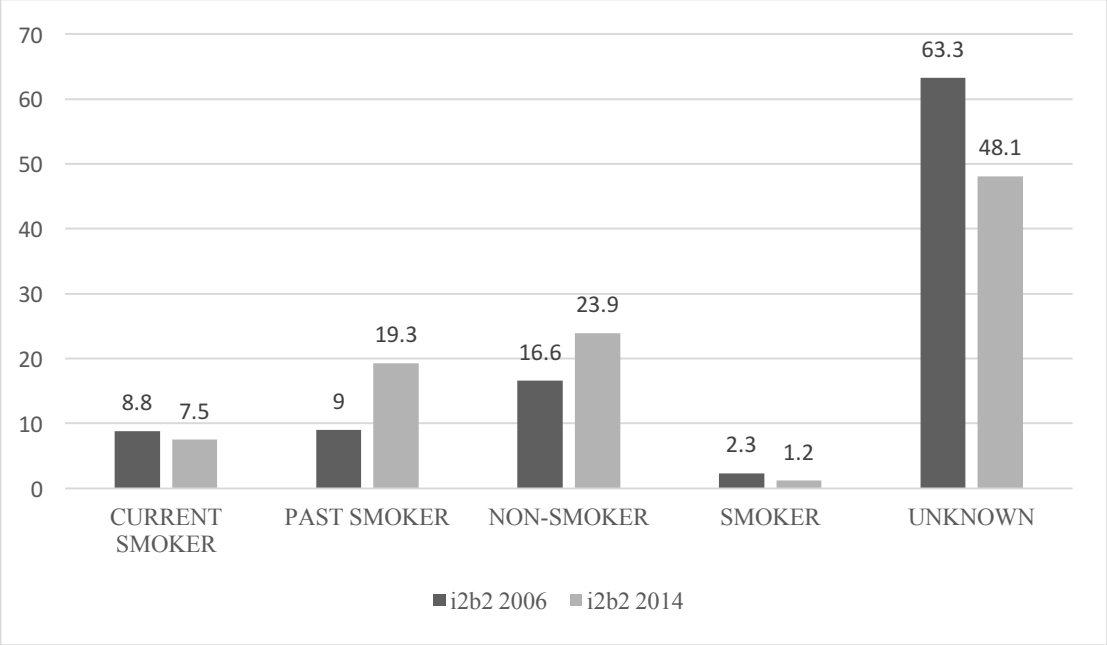

Figure 3.2. A chart showing the relative distribution comparison of the i2b2 2006 and 2014 training data

3.2.3. System Output Specification

The expected output from the system is an extraction of smoking (and alcohol consumption) status for each record. The output should be arranged for ease of comparison with the gold standard.

The expected output format from the system is shown in Figure 3.3. If the dataset contains a gold standard, it should be written on the line next to the prediction of the

related record. Otherwise, the output should contain just the prediction for each record, as shown in Figure 3.4.

```
Line 1: Record ID 1, Prediction
Line 2: Record ID 1, GoldStandard
Line 3: Record ID 2, Prediction
Line 4: Record ID 2, GoldStandard
.
.
.
Line n: Record ID x, Prediction
Line n + 1: Record ID x, GoldStandard
```

Figure 3.3. Expected output format from the system, if the dataset contains a gold standard

```
Line 1: Record ID 1, Prediction
Line 2: Record ID 2, Prediction
Line 3: Record ID 3, Prediction
Line 4: Record ID 4, Prediction
.
.
.
Line n: Record ID x, Prediction
Line n + 1: Record ID x + 1, Prediction
```

Figure 3.4. Expected output format from the system, if the dataset does not contain a gold standard

The specified system output is designed to ease the process of measuring precision, recall, and F-measure for evaluation, as well as to make the error analysis more convenient.

3.3. Overview of the System Workflow

This section discusses the overview of the system workflow that will be developed according to the specifications and requirements in the previous section. The system will have five components: data preparation, pre-processing, information extraction, post-processing, and evaluation.

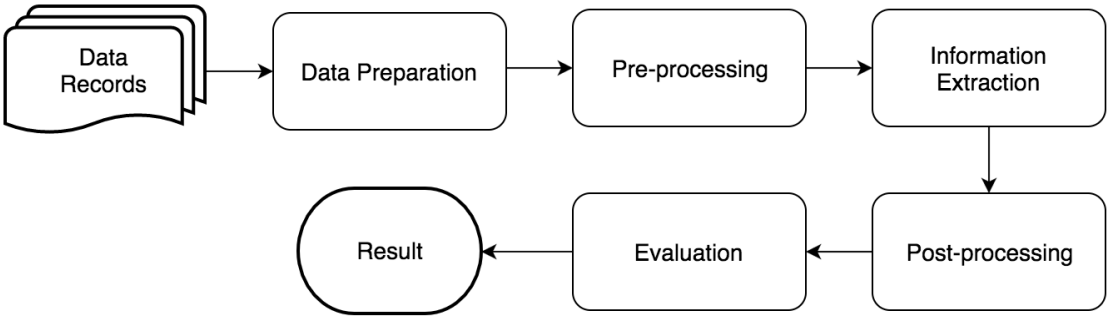

Figure 3.5. Main system workflow

Figure 3.5 outlines the main workflow of the system that will be developed. Each component of the workflow consists of several sub-components to perform specific tasks. All of the components will be integrated to form a full system workflow.

3.3.1. Data Preparation

Two train datasets from the i2b2 2006 and 2014 challenges will be acquired as primary references to develop and evaluate the system. These two datasets have different XML annotations, as they have different purposes in the challenges. The i2b2 2006 dataset, as can be seen in Figure 3.6, has a straightforward annotation that is most suitable for the scope of this project. It lists all the records in a single XML file and attaches the label (gold standard) directly to each record. The important annotations within the data are `<RECORD>` which bounds each document and holds a document ID, `<SMOKING>` which contains a gold standard, and `<TEXT>` which stores a narrative.

```
<ROOT>

    <RECORD ID="xxx">
        <SMOKING STATUS="CURRENT SMOKER"></SMOKING>
        <TEXT>
            ----- free text about patient's condition -----
        </TEXT>
    </RECORD>

    <RECORD ID="yyy">
        .
        .  -----  record description -----
        .
    </RECORD>

    .
    .   -----  a list of records -----
    .

</ROOT>

```

Figure 3.6. Dataset format of the i2b2 2006 Challenge

In contrast, the i2b2 2014 stores a record in a separate file, and each of them contains a number of annotations of medications, heart disease symptoms, and temporal expressions which are irrelevant to this project.

The sections marked with dashed rectangles in Figure 3.7 indicate the important parts from the i2b2 2014 dataset. The section within the `<TEXT>` tag is the narrative, while the `status` attribute in the outer `<SMOKER>` tag denotes the gold standard for that document. As for record ID, the unique file name of each record will be used.

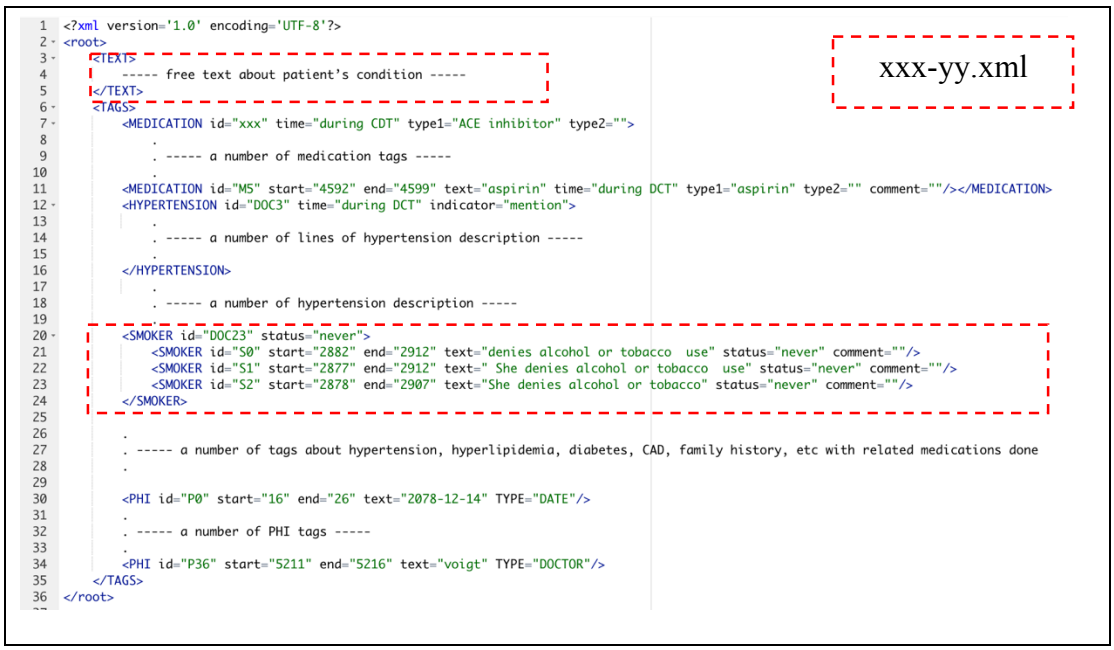

Figure 3.7. Dataset format of the i2b2 2014 Challenge

Thus, a conversion system is needed to make the i2b2 2014 data compatible with the system. The converter should be able to grab important parts in each file of the i2b2 2014 dataset and merge them into a single file identical to the i2b2 2006 format. The illustration of this conversion can be seen in Figure 3.8.

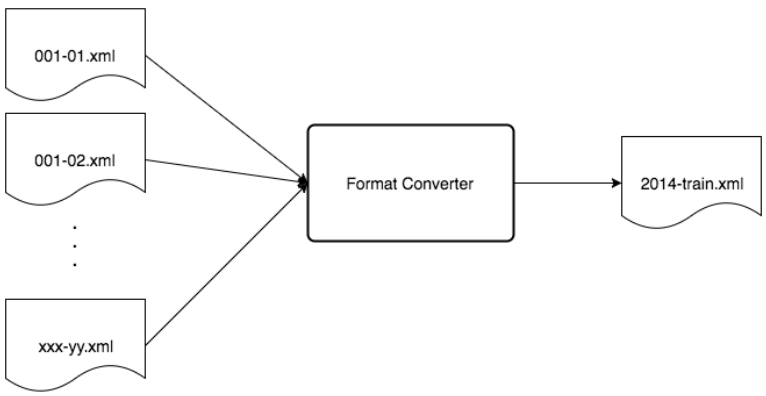

Figure 3.8. Converting data to match the i2b2 2006 format

The i2b2 2014 dataset contains all the required information to support this project, but the tag names are different compared to the i2b2 2006 dataset. Table 3.6 shows the compatible elements in both datasets.

Table 3.6. Component comparison of the i2b2 2006 and 2014 datasets

i2b2 2006 elements (used in this project)	**i2b2 2014 equivalent**	**Note**
<RECORD ID=xxx>	File name	Document number
<SMOKING STATUS=xxx>	<SMOKER status=xxx>	Gold standard
<TEXT>	<TEXT>	Clinical narrative

3.3.2. Pre-processing

Before the data is passed into the main information extraction algorithm, it must be pre-processed to aid in further processing. Because the system that will be developed mainly uses lexical features (e.g. smoking-related keywords, time mentions, and negative contextual cues), two pre-processing steps will be done: tokenisation and sentence splitting.

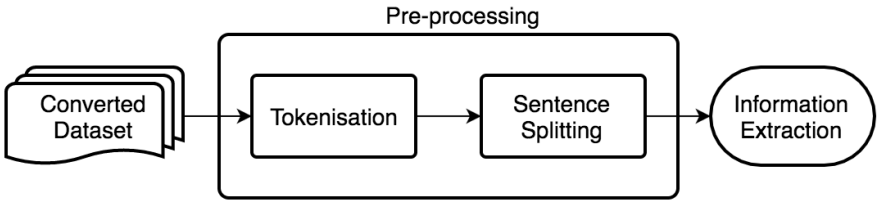

Figure 3.9. The workflow of pre-processing stage

The tokenisation process splits long passages into sets of tokens, the smallest component of a text. The i2b2 2006 data have already been tokenised by the data provider. Each token is delimited by space. Given the condition of the data, re-tokenisation is still needed, since the system to be developed could have different annotation for tokens. The advantage is that the system should need less time to re-tokenise the i2b2 2006 data.

Sentence splitting is needed to indicate the beginning and the end of a sentence. It is important in lexical analysis, because relating tokens in different sentences could

mislead the meaning. Figure 3.10 shows an example of this case. Based on the term "*continues to smoke*", the classification system should group the record into CURRENT SMOKER. However, since there is no proper sentence splitter deployed, the algorithm confuses with the term "*in the past*" in the next sentence, and the record is considered as PAST SMOKER. The i2b2 2006 data has already performed this process. Each line in the record indicates a sentence.

*He **continues to smoke**. **In the past**, patient had been noted to be non-compliant with cardiac medications.*

Figure 3.10. An example of a false extraction as a result of improper sentence splitting

Both the tokenisation and the sentence splitting processes are the vital parts of the workflow. They will supply the required annotations to be processed further by the information extraction algorithm to determine smoking and alcohol consumption status.

3.3.3. Information Extraction

Information extraction is the core component of this project. In this part, the main idea of the algorithm to extract social factors will be discussed. The rules will be crafted manually based on the lexical characteristics of the training data. As this process is critical, the idea explained here is developed based on the successful practices of previous attempts to address similar problems [32] [33] [34]. As a note, this stage only considers the smoking status extraction. The alcohol consumption status extractor will be developed based on the final model of the smoking status extractor by modifying the keywords.

As an overview, the rules are purposed to detect explicit mentions of social factors (e.g. *smoking, smoked, cigar, tobacco*) in each sentence of a record. These mentions are then compared to the surrounding words that indicate the patient's smoking condition (e.g. *past*, *currently*, *former*, *stop*) to form phrase-level prediction, as can be seen in Table 3.7. All of the detected phrase-level predictions in a record will then be considered to form a record-level label in the post-processing stage.

Table 3.7. Examples of how the prediction will be done in phrase-level

Example	Smoking keyword	Surrounding keywords	Prediction
currently smoking a pack per day	smoking	currently	CURRENT SMOKER
tobacco – no	tobacco	no	NON SMOKER
he stopped smoking five years ago	smoking	stopped	PAST SMOKER
he smoked once	smoked	-	SMOKER

To achieve such prediction, several steps are needed to ensure an optimum performance and minimal error rates. Below are the detailed ideas of how this process will be accomplished.

3.3.3.1. Social Factors Keyword Detection

One of the most important parts to detect the presence of social factors noted in a clinical record is the explicit mentions. There are various mentions of smoking status in the record. Some of them are properly mentioned such as *smoking*, *smoke*, *tobacco*, and *cigarette*, while the others are abbreviated like *tob* and *cigs*. The challenge is to detect as many smoking keywords as possible, since if the model fails to detect the main smoking related keywords in a record, the process will stop and the record will be directly classified as UNKNOWN.

3.3.3.2. Crafting the Rules

Once the main smoking keywords have been defined, the next step is to examine the surrounding keywords to predict whether the word or phrase indicates a CURRENT SMOKER, PAST SMOKER, or NON-SMOKER. The SMOKER status will only be given if no information is present regarding the current condition of the patient found near the smoking keywords.

Two types of rules are designed based on how the keywords appear in the records. The first one is frozen expressions, and the other is flexible expressions.

a. Frozen Expressions

This type of rule will be crafted to catch the explicit and obvious mentions of smoking condition in a record. As an instance, if there is a sentence like "*The patient is a current smoker*" in the record, it should be directly classified as CURRENT SMOKER since it is clearly mentioned in the record. The advantage of this rule is to make the classification process efficient and at the same time avoid misclassification if other ambiguous smoking keywords are within the record. It is important that this type of rules be carefully crafted. Only obvious phrases like "*currently smoking*" or "*non-smoker*" should be defined in this type of rule, as it will bypass further processes and will be directly classified as one of the three smoking statuses (CURRENT SMOKER, PAST SMOKER, or NON-SMOKER). Some examples of frozen expressions are shown in Table 3.8.

Table 3.8. Examples of frozen expression

Example	Classification
*He is a **current smoker**.*	CURRENT SMOKER
*The patient is a **non-smoker.***	NON-SMOKER
*Heavy alcohol use, he is also a **past smoker.***	PAST SMOKER

b. Flexible Expressions

Flexible expressions will be executed if smoking keywords are detected in the record, but the frozen expressions failed to catch the possible smoking status. This type of rule is the most difficult task in the information extraction part, since the crafted rules should be able to catch semi-explicit mentions and to tolerate ambiguities to some degree. This is also the last filter of the smoking predictor. If the rules failed to identify the record as a CURRENT SMOKER, PAST SMOKER, or NON-SMOKER, the record will be classified as SMOKER since there is not enough evidence to classify it as one of those three statuses.

Table 3.9 shows some examples of the types of expressions that should be handled by flexible expression rules. It can be seen that various ways can be used to express the smoking condition of a patient. The system should be able to recognise the tense used

in the sentence, either past or present, as the tense may indicate the patient's current condition. Furthermore, the "surrounding keywords" may be located before or after the smoking keywords, separated by several unimportant words.

Table 3.9. Examples of flexible expression

Example	Classification
1) *He **has been a smoker** for the last 50 years* 2) *The patient is a **1 1/2 pack per day smoker**.*	CURRENT SMOKER
1) ***Smoking**: **No** history of **cigarette** use.* 2) ***Denies** use of alcohol and **tobacco**.*	NON-SMOKER
1) *Notable for heavy alcohol use and history of heavy **tobacco use**, although he **quit** two years ago.* 2) *The patient **smoked** one pack of cigarettes per day **from age 16 to 50** .*	PAST SMOKER

The tolerance of the gap between the smoking keyword and its surrounding keywords should be considered as well. Some important information will not be captured if the gap tolerance is too short. However, if the gap tolerance is too long, a false match may occur, as can be seen in Table 3.10.

Table 3.10. The importance to set a proper gap tolerance between keywords

Example	Remark
***tobacco** 1 ppd x 35 years, began 92, **quit** 94*	The gap between "tobacco" and "quit" is 7 words. If the gap tolerance is too short, it may be falsely classified as SMOKER instead of PAST SMOKER.
*She **does not** consult to physician and in recent days she is **smoking***	If the gap tolerance is too long, it is possible to mark this sentence as NON-SMOKER instead of CURRENT SMOKER.

The decision of how much gap tolerance should be given for each rule will be made based on the cases in the available training data. The implementation of this concept is given in section 5.4 about rules development.

3.3.3.3. Error Prevention

Error prevention holds an important role in this information extraction stage. As the clinical records are in a free-form text format, the physician can write almost anything in any forms on the clinical records. A false match may occur due to context misinterpretation as can be seen in Table 3.11.

Two main causes of misinterpretation were detected in the training set. The first is because of the family history mentions in the clinical record. This is a common cause since sometimes the condition of the patient's health is affected by the surrounding conditions, including his/her family history. The second cause is because of the suggestions for the patient written by the physician in the record. The suggestion often contains phrases or sentences which are not the current condition of the patient, like "*please attempt to stop smoking*" or "*please try to eat healthy food*". If the rules failed to recognise that it was just a suggestion, a false prediction may occur.

Table 3.11. Misclassification due to context misinterpretation

Example	Remark
***His father** is a **heavy smoker**.*	Based on the sentence, the classifier could label it as CURRENT SMOKER. But in reality, the context of that sentence is explaining the condition of the patient's father, not the patient him/herself.
1) ***Suggestion**: please **stop smoking** to make the medication works.* OR 2) ***Totally must quit** all **cigars**.*	This sentence is just a suggestion for the patient. It can be inferred implicitly that the patient is a CURRENT SMOKER. But, if the rules do not consider the context, it may be classified as PAST SMOKER.

Aside from context misinterpretation, error may also be caused by the negative words around the main keywords. Some examples are shown in Table 3.12.

Table 3.12. Misclassification due to negative words

Example	Remark
*He continued to see Dr Franklin frequently but has **not quit smoking**.*	The word "not" which precedes "quit smoking" must be taken into account when considering the label for this sentence.
*The patient **denies** being a **current smoker**.*	The word "denies" negates the phrase "current smoker".

While the ways in which smoking conditions are described in the training data vary significantly, and it is impossible to capture all the variations, the concept here was designed to tolerate ambiguities at a certain level. Thus, the design is expected to catch major variations of smoking mentions in clinical narratives.

3.3.4. Post-processing

The post-processing stage is meant to identify an appropriate record-level prediction based on the phrase-level predictions in a record. This process is tricky because there is a small chance that a record contains more than one phrase-level prediction. In this condition, prioritisation is needed to determine the likelihood of a label to be correct compared to the other labels.

Table 3.13 shows some examples of records that contain more than one phrase-level prediction. A deep analysis by comparing the output from the system and the gold standard is needed to determine an appropriate precedence level for each smoking status label. This issue will be discussed in the development chapter.

Table 3.13. Examples of record with multiple phrase-level predictions

Entry in record	Phrase-level predictions	Record-level prediction
*Now she **does not smoke** and has no history of coronary artery disease.* *The patient was a heavy **smoker in the past**.*	NON-SMOKER PAST SMOKER	PAST SMOKER
*Condition: **currently smoking*** *He **does not smoke** when he was child, yet he has a lung disease since then.*	CURRENT SMOKER NON-SMOKER	CURRENT SMOKER

3.3.5. Evaluation Metrics

The record-level prediction from the system will be evaluated by comparing it with a gold standard. The comparison will result in a confusion matrix with four entities: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). The components in the confusion matrix are explained in Table 3.14.

Table 3.14. Explanation of the confusion matrix components

Confusion matrix component	Explanation
True Positive (TP)	Positive records that were correctly classified by the system
False Positive (FP)	Negative records that were incorrectly classified by the system
True Negative (TN)	Negative records that were correctly classified by the system
False Negative (FN)	Positive records that were incorrectly classified by the system

These confusion matrix components will then be used to measure precision (how many selected records are relevant) and recall (how many relevant records are

selected) for each smoking and alcohol consumption category. The next step is to calculate the harmonic mean of precision and recall for each classification category using F-measure. The formulas to evaluate the performance of the generated model are shown below.

$$P = \frac{TP}{TP + FP}$$

Equation 3.1. Precision formula

$$R = \frac{TP}{TP + FN}$$

Equation 3.2. Recall formula

$$F = \frac{(1 + \beta^2) \times P \times R}{(\beta^2 \times P) + R}$$

Equation 3.3. F-measure formula

The parameter β is used to configure the importance of precision and recall. For this research, $\beta = 1$ will be used to set equal weight of precision and recall [6].

The overall score of the system will be calculated by combining scores from each smoking or alcohol consumption category in two ways. The first method uses the macroaveraged measurement, by considering the same weight for all five smoking or alcohol consumption statuses regardless of the imbalanced condition of the dataset. This measurement is used to make sure that the proposed solution is robust and able to recognise any smoking and alcohol consumption status well. The second one, microaveraged measurement, puts equal weight on each document in the dataset. It simulates the real-world performance of the developed system. The parameter *M* in both macroaveraged and microaveraged F-measures denotes the number of categories in the data.

$$F(macro) = \frac{\sum_{i=1}^{M} F_i}{M}$$

Equation 3.4. Macroaveraged F-measure formula

$$F(micro) = \sum_{i=1}^{M} \frac{F_i(FP_i + FN_i)}{(TP + FP)}$$

Equation 3.5. Microaveraged F-measure formula

As a note, if "performance" or "score" is mentioned in some sections of this document, it refers to the F-measure calculations explained previously.

4. System Design

The system design chapter discusses the architecture of the system based on the research methodology in the previous chapter. First, the system components will be outlined. The components will then be arranged in diagrams that show the main workflow and interactions between the components.

4.1. System Components

Figure 3.5 in the previous chapter shows the five important components in this project. Each component consists of several sub-components which will be developed using specific technology platforms. An overview of these technology platforms is provided here. A detailed discussion of how these technologies work can be seen in the implementation chapter (see chapter 5).

1) Data Preparation

The datasets from the i2b2 repository were in the XML file format. To standardise the format for the system's input, a data converter will be developed using the Java programming language. Java is chosen as it has a rich library to manipulate XML files.

2) Pre-processing

After the data has been converted into a suitable format, it will be passed through the pre-processing stage. This stage will be accomplished within the GATE platform. It has numerous processing resources related to text engineering tasks. This pre-processing stage will be done using the English tokeniser and sentence splitter from ANNIE (A Nearly-New Information Extraction System), one of the plugins distributed within GATE.

3) Information Extraction

The information extraction stage will also be conducted within the the GATE platform. The frozen expressions will be captured first using the ANNIE gazetteer. Subsequently, the smoking prediction rules will be developed using JAPE (Java Annotation Pattern Engine). JAPE is a language which is designed to provide annotations over a document based on regular expressions.

4) Post-processing

The post-processing stage will be developed using JAPE as well. Phrase-level predictions from the previous stage will be considered to become a final record-level label in this stage.

5) Evaluation

Java will be used to develop evaluation code. A CSV file output from GATE which contains a prediction and label for each record will be expected. The code will calculate the accuracy, precision, recall, and F-measure (both microaveraged and macroaveraged) for each dataset based on the output from GATE.

4.2. Data Flow and System Boundaries

This section explains the interaction between components within the system using the Data Flow Diagram (DFD). The diagram shows the sequence of the data processing stages as well as the boundaries that exist within the system.

First, the data from the i2b2 online repository will be downloaded in an XML format. The data will then be passed into the data preparation module (stage 1) to be formatted to fit the requirement of the system. This step is crucial, as the non-compatible annotations in the data could disrupt further processes. This stage, and all further stages, will be conducted offline to ensure the confidentiality of the data as this is a part of the Data Usage Agreement from the i2b2 (see section 5.1 regarding the Data Use and Confidentiality Agreement).

Second, neatly formatted data from the previous stage will be taken as an input for GATE. Several processes will be done within the GATE environment. The first process will be pre-processing (stage 2) using the ANNIE modules of English tokeniser and sentence splitter. In this stage, the unstructured text will be parsed into tokens. The beginning and the end of each sentence will also be marked.

Third, the tokenised text will be acquired by the information extraction module (stage 3). This module works primarily by manipulating the value of a token and its surroundings. The ANNIE gazetteer will look for the words or phrases that exactly match its list. Furthermore, the JAPE rules will trace the sequence of tokens and

annotate them based on the instruction given. The goal of this step is to make phrase-level predictions based on the lexical characteristics of the text.

Fourth, the JAPE rules in the post-processing module (stage 4) will give a final label for each record. If gazetteer annotations are found in a record, other phrase-level predictions will be ignored and the classification will only be done based on the gazetteer as the frozen expressions. Otherwise, the phrase-level predictions will be accounted to form a final prediction.

Finally, a CSV output containing the prediction for each record will be taken from the GATE to be evaluated (stage 5) using Java code. The outputs will be compared with a gold standard to calculate precision, recall, and F-measure for each category. Those calculations will then be combined to obtain microaveraged and macroaveraged F-measure as an overall dataset score.

This flow can be seen in Figure 4.1.

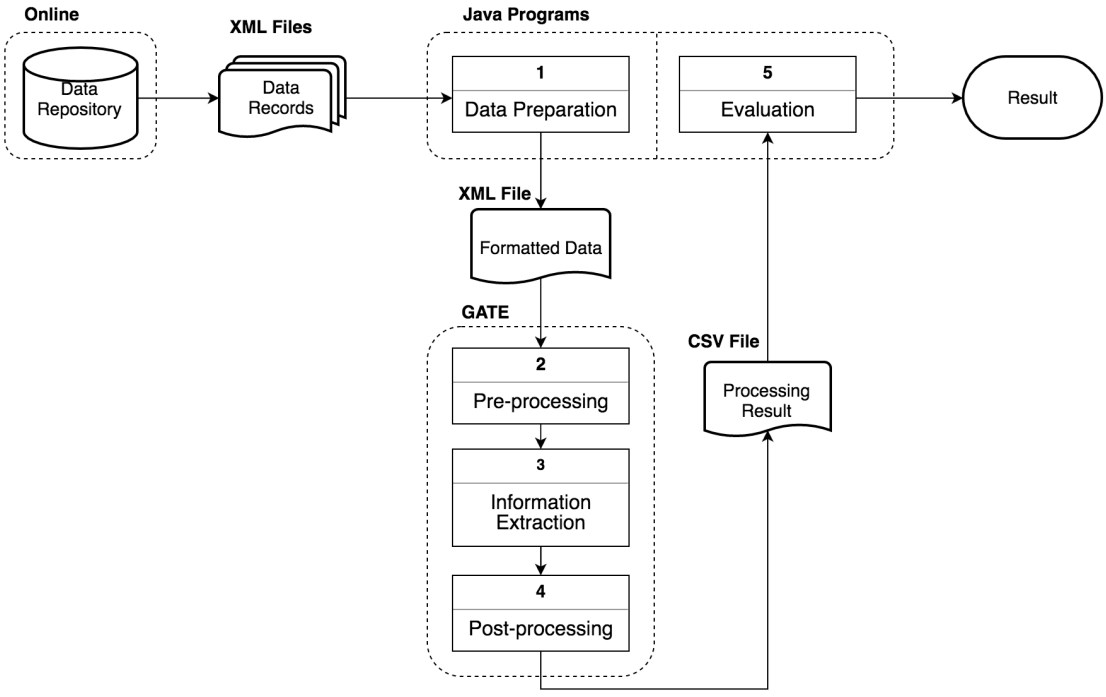

Figure 4.1. The Data Flow Diagram

4.3. System Workflow

After a DFD as an outline of the system components and boundaries has been designed, the next step is to construct a system workflow. This section discusses how the data will be processed through the system from the preparation to the evaluation

phase in higher detail. An activity diagram will be used to define important system elements along with their roles and positions. The diagram will also mark the critical decisions that should be made within the workflow that impact the further processes in the workflow. Unless stated otherwise in the component, the workflow was designed at the record level, meaning that each record in the dataset will run through a complete set of the workflow. The diagram can be seen in Figure 4.2.

4.3.1. Data Preparation

The first process is data preparation. The datasets that have been acquired from the i2b2 repository should be converted before they are passed into the main information extraction algorithm. For the development purpose, each record should have a gold standard annotation attached to it, so the result can be evaluated. All of the i2b2 2006 data have this annotation, as it is the main purpose of the challenge. Meanwhile, a small number of records in the i2b2 2014 corpus do not contain such annotation due to lack of consensus by the annotators. These records will be removed from the system. The rest of the records will be combined to form a single XML file with a specific format (see section 3.3.1).

4.3.2. Pre-processing

The prepared datasets will be passed into the pre-processing mechanism. The narratives inside the <TEXT> tag in each record will be split into tokens and sentences. The ANNIE tokeniser will be used for the tokenisation process. It splits the text into tokens. Five types of token are defined by ANNIE, and are shown in Table 4.1 [35].

The English version of ANNIE tokeniser will be used. It compiles the rules of ANNIE tokeniser with an additional English specific part-of-speech tagger. For example, the tokeniser will join these kinds of constructs in one token: *'30s*, *Cause*, *'em*, *'N*, *'s*, *'d*, *'ll*, *'m*, *'til*, *'ve*. It will also convert negative constructs such as "*don't*" into two tokens ("*do*" and "*n't*") instead of three tokens ("*don*", " ' ", and "*t*") [35].

Aside of tokenisation, the text will be processed with ANNIE sentence splitter as well. The splitter uses a gazetteer to distinguish the function of a period, whether it is intended to mark a sentence full-stop or not (e.g. to mark abbreviations such as Dr.

and *et al.*). Each detected sentence will be marked as `Sentence` and the break between two sentences will be given a `Split` annotation. The sentence splitter is a domain-specific application, because each knowledge domain could have different writing styles.

Table 4.1. The token types defined by ANNIE [35]

Token type	Description
Word	Any set of contiguous letters, including a hyphen (-)
Number	Any combination of consecutive digits
Symbol	Two types of symbol were defined: 1) currency symbol (e.g. '$', '£') 2) normal symbol (e.g. '&', '*')
Punctuation	Three types of punctuation were defined: 1) start_punctuation – e.g. '(' 2) end_punctuation – e.g. ')' 3) other punctuation –e.g. ':' Each punctuation symbol is a separate token
SpaceToken	Two types of SpaceToken were defined according to whether they are pure space tokens or control characters: 1) Space: normal white space in any form (e.g. space, tab) 2) Control: a character that is not printable but initiates a particular action Any contiguous set of space is defined as a SpaceToken

For this project, a default version of ANNIE sentence splitter will be used. One of the distinguishing features between the versions is the way in which the algorithm handles the newline breaks. In the default version, text in different lines will be considered as different sentences, even though there are no sentence-stopping marks (e.g. period, question mark, exclamation mark) detected between them. This is the most suitable version for this project since the physicians often write a short note in a line without "stopping" the sentence properly. If the text in the adjacent line is

considered as a same sentence, it could result in a false marking that may damage the extraction process.

4.3.3. Information Extraction and Post-processing

The next phase after the text has been pre-processed is to extract smoking status from the records. The first approach will be done using the ANNIE gazetteer to detect frozen expressions and to eliminate the sentences that have family mentions within them to prevent errors (see section 3.3.3.3 for the detailed description of error prevention).

The gazetteer is used to identify entity names in the record based on user-defined lists [35]. If a word sequence in the text matches the lists, an annotation will be made to the sequence. Four gazetteer lists will be developed for this project. The detailed description of the lists can be seen in Table 4.2.

Table 4.2. Gazetteer lists to be developed

List name	Purpose	Description
FAMILY list	Error prevention	Contains the keywords related to family names. If the list fired in a sentence, the smoking status detection will ignore that sentence.
CURRENT SMOKER list	Frozen expressions	Contains the frozen expressions of CURRENT SMOKER.
PAST SMOKER list	Frozen expressions	Contains the frozen expressions of PAST-SMOKER.
NON-SMOKER list	Frozen expressions	Contains the frozen expressions of NON-SMOKER.

If at least one of the frozen expression lists fired in a record, further inspection will be done to detect the presence of negative keywords (e.g. *no*, *non*, *not*) that precedes the expression. The presence of such keywords will negate the expression; thus, the frozen expression will be considered invalid. If that condition happens, the detection process will continue further. Otherwise, it will be directly passed into the post-processing stage to be classified as CURRENT SMOKER, PAST SMOKER, or

NON-SMOKER depending on which frozen expression was detected. If more than one frozen expression lists fired, the smoking status with higher precedence will be given. The decision about which status has higher precedence than others will be presented in the implementation chapter.

The next phase, if the frozen expressions failed to catch the smoking status, is the detection using the crafted JAPE rules. The first JAPE rule is purposed to detect the presence of smoking-related keywords (e.g. *smoke*, *cigar*, *tobacco*) in the record. If the rule cannot find such keywords, it is assumed that the record contains no discussion about the patient's smoking condition, and thus it is classified as UNKNOWN.

If the smoking keywords are detected, the next phase is to detect the keywords that indicate PAST SMOKER (e.g. *past*, *former*, *stop*) and NON-SMOKER (e.g. *deny*, *never*) around the smoking keywords. The CURRENT SMOKER keywords in a sentence will be detected later if neither PAST SMOKER nor NON-SMOKER keywords fired in that sentence. This decision was taken since the investigation through the available datasets concludes that the records labelled with CURRENT SMOKER often contain only general smoking keywords without any explicit indication that the patient is currently smoking.

As an instance, a record that just mentions "*condition: smoking*" was classified as CURRENT SMOKER. Thus, the keyword "*smoking*" will be inserted into CURRENT SMOKER rules even though there is no explicit mention that indicates the smoking condition of the patient (e.g. *currently*, *not stopping*). Therefore, to prevent false double classifications of the sequences similar to "*stop smoking*" or "*smoking in the past*", the CURRENT SMOKER status will only be given if there are no PAST SMOKER or NON-SMOKER rules detected in the sentence.

Following the smoking status detection is the post-processing stage, which annotates record-level prediction based on the phrase-level predictions within it. If no CURRENT SMOKER, PAST SMOKER, or NON SMOKER annotations are found in the record, it will be classified as SMOKER. Otherwise, one of the three statuses will be given to the record based on the smoking status precedence.

4.3.4. Evaluation

This is the last phase of the workflow. The output from the previous phase will be taken out as a CSV file and then compared to a gold standard. The evaluation metrics such as precision, recall, and F-measure will be used to calculate the performance of the developed system. The errors will be analysed for further improvements.

4.4. System Interaction

After the system components, boundaries, and flow have been defined, the next step is to design how the components interact to each other. This definition will give insights about the role of each component in the process. The sequence diagram, which can be seen in Figure 4.3, is used to illustrate the system interaction.

Ten system components were defined in the sequence diagram, as shown in Table 4.3. Some differences in the order of components are seen between the activity diagram and the sequence diagram. These differences are due to the technical modularity of the development environment. The activity diagram was designed for ease of understanding the system flow. Conversely, the sequence diagram was constructed to represent the real development environment condition. Aside from the differences, both diagrams represent the same concept but in a different abstraction level.

Table 4.3. The components in the sequence diagram

Component	Description
Data Converter	Converts the data into a suitable format
ANNIE English Tokeniser	Tokenises the text
ANNIE Sentence Splitter	Annotates the beginning and the end of the sentences, annotates the gap between two sentences.
ANNIE Gazetteer	Lists the frozen expressions and prevents errors
JAPE Smoking Keywords Detector	Detects smoking related keywords
JAPE Past Smoker Detector	Detects past smoker keywords

JAPE Non-Smoker Detector	Detects non-smoker keywords
JAPE Current Smoker Detector	Detects current smoker keywords
JAPE Record-Level Predictor	Decides record-level predictions based on gazetteer or JAPE rules
Evaluator	Compares the result to the gold standard

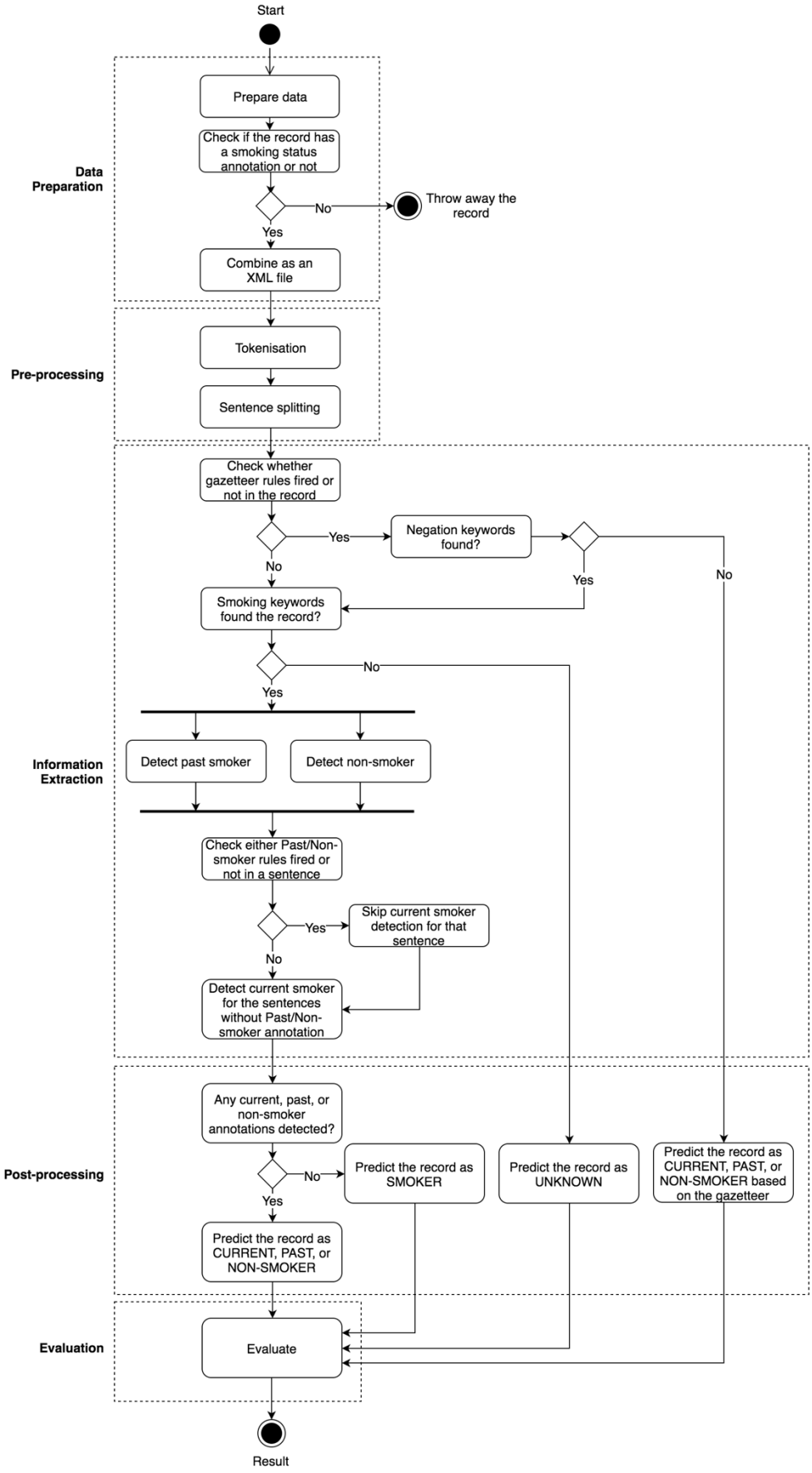

Figure 4.2. The activity diagram of the system

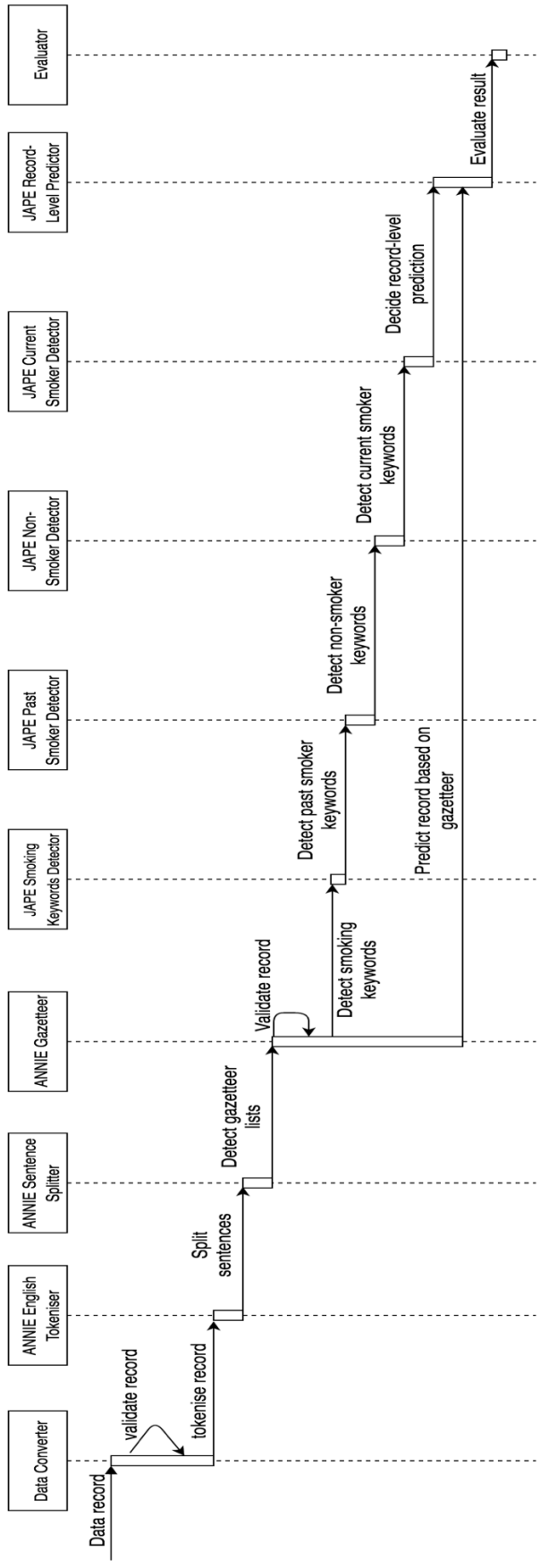

Figure 4.3. The sequence diagram of the system

5. Implementation

This chapter explains how the research methodology was implemented in the project. First, the data use agreement from the i2b2 was signed to obtain the required data. Then, the development environment was investigated. The next part included developing the system in four iterations and evaluating the results by using training data. Last, the final version of the smoking predictor was repurposed to predict alcohol consumption status.

5.1. Data Use and Confidentiality Agreement

Because this project will use sensitive data from the i2b2 repository, an approval scheme was needed to acquire the data. A regulation has been defined by the i2b2 for those who want to use the data from its repository [36]. It was stated as a terms and conditions in a Data Use and Confidentiality Agreement that should be signed be the applicants. As a summary, the agreement mentions that the applicant should agree that the data will only be used for the specific purposes stated in the agreement. The document also states that it is the responsibility of the applicant to prevent the disclosure or illegal use of the data.

In this regard, the document was studied, signed, and sent back to the i2b2 on behalf of the researcher of this project on March 14, 2016. The approval was granted a day later, along with an internet account to access the data.

In addition to that agreement, an ethical approval with an id number of 263 was submitted to the University of Manchester, and has been approved by the institution. The statements in the approval form include the description of the research, the data that will be used, and the participants of the research.

All of the clinical data on the i2b2 repository was anonymised prior to being published. This means that all of the possible sequences that contain identifiable features (e.g. patient name, physician name, phone number, temporal expressions, address, etc.) were replaced with dummy content to protect the privacy of the patients.

5.2. Development Environment

This section gives an overview of the General Architecture for Text Engineering (GATE) as the main environment that was used to develop the system and the components within it: A Nearly-New Information Extraction System (ANNIE) and the Java Annotation Pattern Engine (JAPE) plugins.

5.2.1. General Architecture for Text Engineering

The General Architecture for Text Engineering is a GNU-licensed open-source software which has been developed using Java. It is intended to solve major text processing tasks. The latest release per August 2016 was Gate-8.2, which was used as a development environment for this project. GATE contains many plugins that can support a diverse assortment of text mining tasks. The system is differentiated into two versions: the GATE Developer version which is supported by an intuitive Graphical User Interface (GUI) and the GATE Embedded version which is basically a command line version that can be integrated with other systems. This project was developed using GATE Developer.

Figure 5.1 shows the main application window of GATE Developer. Users can employ the drag-and-drop feature to design their text mining workflow, assign suitable parameters for each processing component, and choose the corpus to be processed. Once the workflow is ready, the users can click the “Run this Application” button to execute it.

After the workflow execution has been completed, the users can see the generated annotations by clicking the dataset that they have selected before and activate the “Annotation Sets” and “Annotations List” tabs above the main text area. Figure 5.2 shows that there are only two annotations selected to be displayed: “SmokeMention” and “SmokingIndicator”. The annotations list that appears below the main text area can be processed further by exporting the text into a CSV file. The comprehensive guide to use the GATE platform can be seen in its documentation page [37].

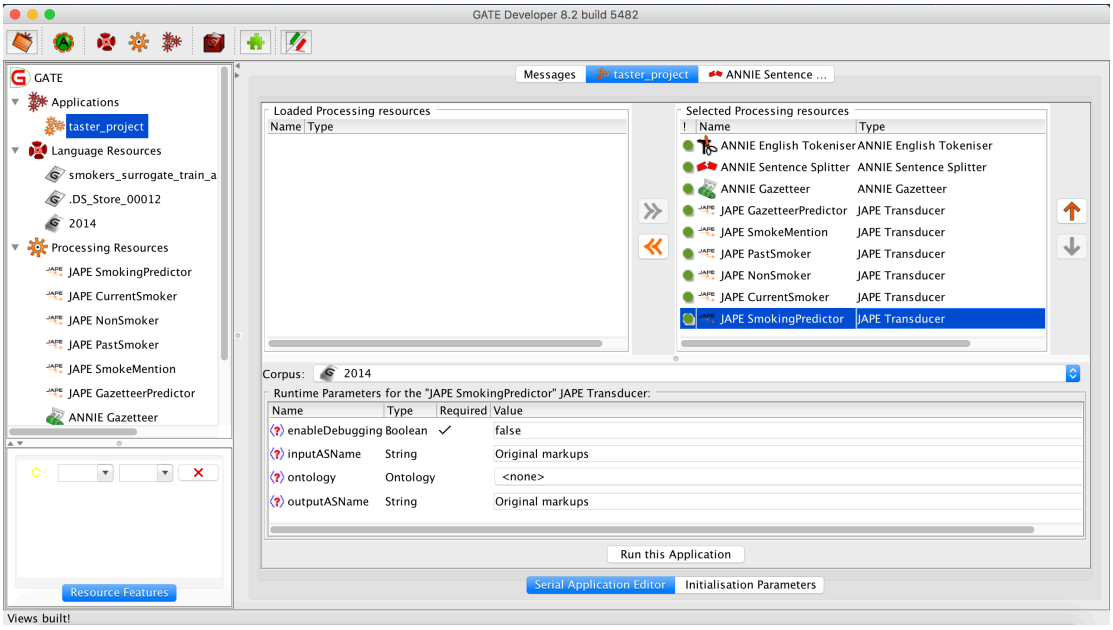

Figure 5.1. The main application window of GATE Developer

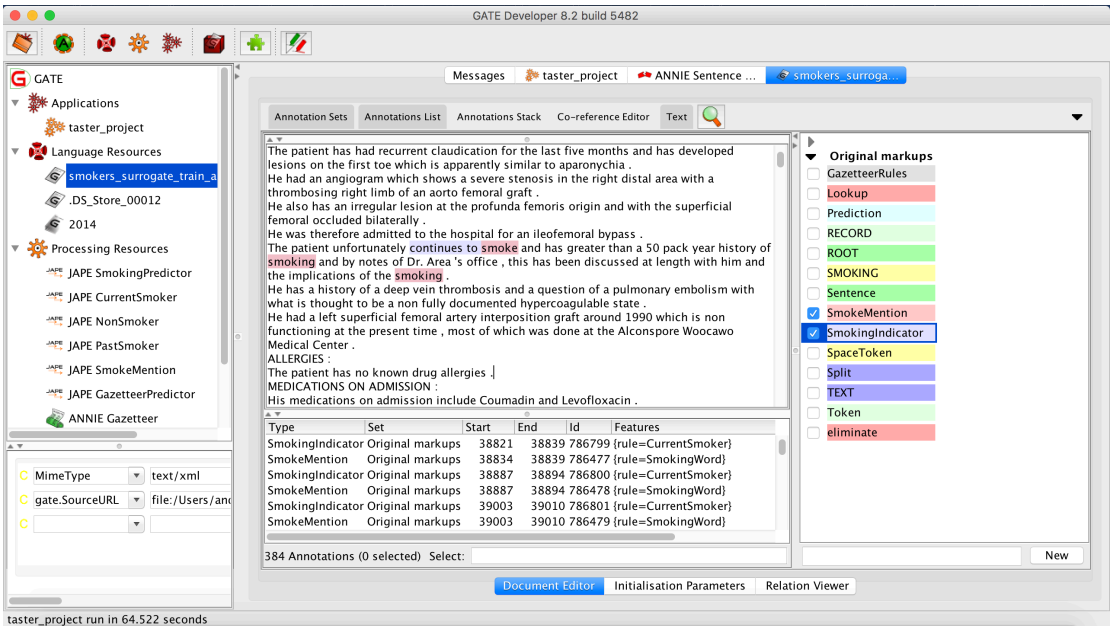

Figure 5.2. The annotation set view of GATE Developer

5.2.2. Java Annotation Pattern Engine

The Java Annotation Pattern Engine is a language to annotate text based on regular expressions (regex). The language consists of a set of phases. Each phase contains a set of instructions/rules. The phases are executed sequentially to support each other. One JAPE phase is stored in a file with a .jape extension.

A JAPE phase consists of two sides: The Left Hand Side (LHS) and the Right Hand Side (RHS) rules. The rules in the LHS are meant to describe the pattern in the text to be matched, while the codes in the RHS consists of a manipulation statement that should be executed if the rules in the LHS fired.

Consider an example of a JAPE phase in Figure 5.3. A JAPE phase begins with a header to define the phase name, the input needed to be processed in the rules, and the execution options. The example defines `control=appelt` as an option. The appelt mode means that if a rule fired, any rules after that will not be executed. In this case, if `Rule #1` is fired, `Rule #2` (and rules after that, if any) will not be examined.

```
Phase: smokingpredictor
Input: RECORD PastSmoker CurrentSmoker
Options: control=appelt

// Rule #1
Rule: PastSmokerRule
(
{RECORD contains PastSmoker}
):predictor
-->
:predictor.Summary = {PREDICTION = "PAST SMOKER"}

// Rule #2
Rule: CurrentSmokerRule
(
{RECORD contains CurrentSmoker}
):predictor
-->
:predictor.Summary = {PREDICTION = "CURRENT SMOKER"}
```

Figure 5.3. An example of a JAPE phase

Each rule has its own LHS and RHS. The LHS is the text that precedes the `-->` sign, while the RHS is the opposite side of it. As an instance, the LHS of the first rule is meant to detect if the record has a `PastSmoker` annotation on it. If the LHS rules fired, a new annotation type named `Summary` will be added to the record. It will contain an attribute named `PREDICTION` with `PAST SMOKER` as its value.

5.2.3. A Nearly-New Information Extraction System

GATE comes with an information extraction system named A Nearly-New Information Extraction System (ANNIE). ANNIE was developed using finite state algorithms and JAPE rules. It provides various information extraction tasks that can be arranged as pipeline components within GATE. Figure 5.4 illustrates the roles of ANNIE within GATE, along with another system named LaSIE (Large Scale Information Extraction) [38].

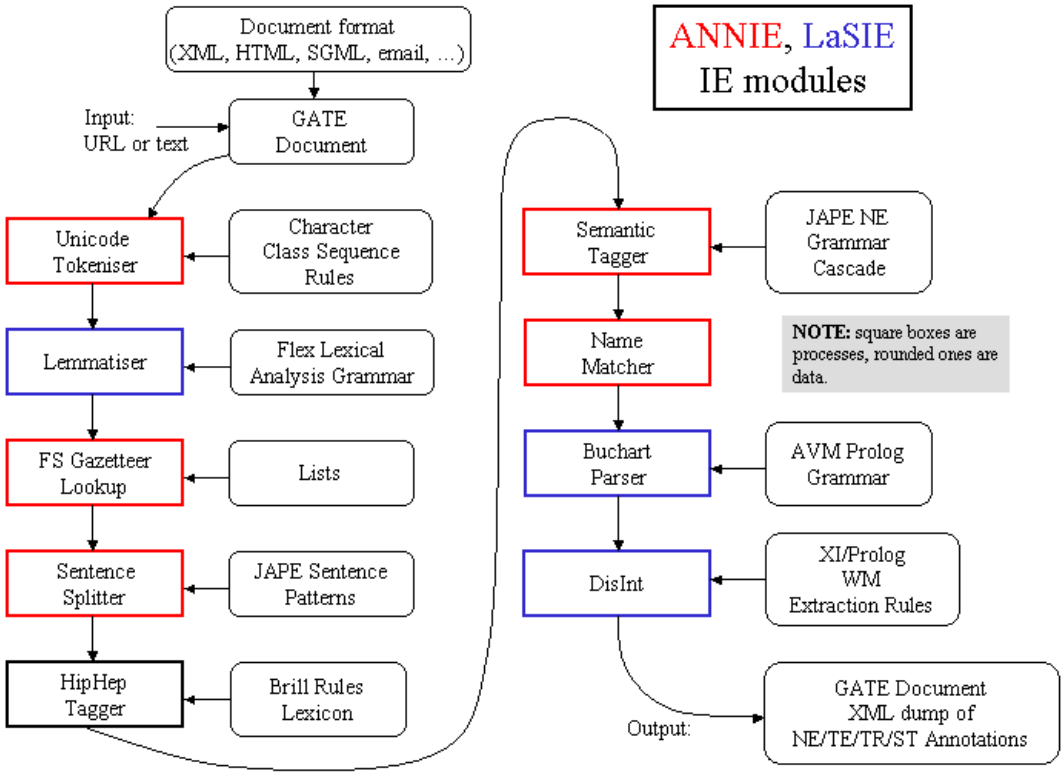

Figure 5.4. The roles of ANNIE (red rectangles) within the GATE system [38]

This project used three ANNIE components: English tokeniser, sentence splitter, and gazetteer. The detailed explanation of how English tokeniser and sentence splitter works can be seen in section 4.3.2 about pre-processing mechanism, while the details of the gazetteer can be seen in the section subsequent to that.

5.3. Development Model

The system was developed by adopting the spiral model. It is a modification of the conventional waterfall model by adding cycles in the development process. Similar to

the waterfall model, the spiral model has four stages (objectives, design, development, test) in one iteration [39]. The iterative paradigm in the spiral model helps the developer to mitigate the risks as early as possible by delivering a system prototype in each iteration to be evaluated [40].

Four development cycles have been performed in this project. Each cycle has specific purposes of development and utilises different train datasets. The detailed description of the purposes and mechanisms in each iteration can be seen in the following sections.

5.3.1. Iteration 1: Taster Project

The first iteration of the development aimed to test out the feasibility of the project and to deliver a taster prototype. A total of 398 records from the i2b2 2006 dataset were used in this iteration. In this phase, the keywords that indicate smoking mentions were identified. Initially, three kinds of smoking mention were identified: *smok*, *cigar*, and *tobac*. Tokens that contain a substring from any one of the three identified mentions were marked as smoking keywords.

The next step was inspecting the surrounding words from the detected smoking keyword to specify the phrase-level smoking category. Some rules to adjust negative keywords and double negations were made to prevent false matching. In addition, the precedence of each smoking status in this iteration was made purely based on manual inspection to the training set.

The deployed workflow was debugged several times using the same training data. The errors found were corrected. The final result of the taster project was delivered at the end of June 2016. This taster version provides a solid foundation of the workflow to be improved in further iterations.

5.3.2. Iteration 2: Modifying the Rules Using the i2b2 2014 Dataset

The second iteration of the project was meant to enhance the taster version. The main concern in this iteration was extending the training dataset by 771 records of the i2b2 2014 data. This phase was done by executing the taster version using the new dataset. As the new dataset almost doubled the size of the i2b2 2006 data, a number of new errors were found. All errors were documented and analysed to improve the system.

An analysis showed a number of new smoking keywords found in the new dataset, which were not detected by the taster system. It also resulted in various new smoking expressions in the record. The rules were revised accordingly.

Another feature that was added in this iteration was an additional rule to the CURRENT SMOKER phase. If either PAST SMOKER or NON-SMOKER rules fired in a sentence, the CURRENT SMOKER checking will not be done in that sentence. This new rule helps to prevent false matching to some degree when smoking status keywords with high generality appear in a record (see section 4.3.3 about information extraction and post-processing for the detailed explanation).

5.3.3. Iteration 3: Adjusting the Rules between Two Datasets

The next iteration focused on adjustment between the two datasets. As the rules were significantly modified based on the i2b2 2014 data, the performance for the i2b2 2006 dataset was degraded. A couple of test-and-evaluate iterations were made to balance the system performance between these two datasets. Some additional rules were made to handle special cases in each dataset. This balancing mechanism improved the performance of the system because it handled greater variations in more specific ways.

Conversely, some technical issues were found in this iteration. It was recognised that the two datasets have different record writing formats. The i2b2 2006 dataset separates sentences in different lines, meaning that each line contains one sentence. However, the i2b2 2014 dataset has more inconsistencies. Line breaks occur in between sentences. These differences resulted in some inaccuracies as a result of keyword mismatching and rules that overflow a sentence. These issues were documented and resolved in the next iteration.

5.3.4. Iteration 4: Developing Gazetteer Entries and Preparing Alcohol Consumption Status Prediction

The last iteration of the development was aimed to polish the system and repurpose it to predict alcohol consumption status. The technical issues in the previous iteration were addressed by adding additional parameters to prevent the rules to overflow the sentence boundary. A couple of rules were added to prevent the rules from detecting

an unintended smoking condition (i.e. marijuana smoking) which was not the part of this project. In addition, a number of gazetteer entries were created and embedded in the system to detect the frozen expressions. The gazetteer was also meant to prevent misclassification as a result of context misinterpretation (see section 3.3.3.3).

To finish the iteration, a method to decide the smoking status precedence was formulated. A matrix was introduced to compare the total number of phrase-level predictions detected in the datasets against the number of records for each smoking category.

After the development and the trials using training were completed, the last phase was to modify the the system to predict alcohol consumption status. This was accomplished by altering the set of keywords in the rules to the alcohol-related keywords. No other adjustments in the rules were made as the project assumes that the alcohol consumption status in the dataset is expressed in similar ways to the smoking status.

The summary of the four iterations can be seen in Table 5.1. The detailed discussion about the rules development, alcohol consumption status prediction, and evaluation will be explained in the next sections.

Table 5.1. Summary of the development iterations

Iteration	Datasets used	Purpose
Iteration 1	i2b2 2006	Taster prototype
Iteration 2	i2b2 2014	Extend the rules by examining another dataset
Iteration 3	i2b2 2006, i2b2 2014	Balancing the rules
Iteration 4	i2b2 2006, i2b2 2014	Final touch and modify the rules to predict alcohol consumption status

5.4. Rules Development

This section explains the final rules that have been forged through the four iterations explained previously. The rules for each smoking status are illustrated in graphs followed by an explanation and examples. The index tables for the terms used in the

graphs can be found in Table 5.2 to Table 5.6 following the illustrations. The rules were created based on the activity diagram of this project (see Figure 4.2).

5.4.1. Current Smoker Rules

CURRENT SMOKER is among the status with various mentions, yet the rules crafted were the simplest compared to the other rules. When making the rules, one of the things that should be considered is the generality. The rule should not be too specific, as it may over-fit the training set and is not good for the unseen data. Many variations of CURRENT SMOKER mention cannot be generalised as they are only appeared in one or two records. Only patterns with a certain level of generality should be considered for rules.

Figure 5.5 illustrates the rules to detect CURRENT SMOKER. The boxes indicate the suitable keywords group that should exist to fire the rules. The "max *n* tokens" in the arrows indicate that the distance between the left box and the right box is *n* tokens at maximum, while the note "*m* tokens back" means that the *m* tokens before the right box should match the condition. As an instance, the first rule in Figure 5.5 means that the rule will fire if there is a keyword in gaz_current group mentioned in the text and there are no keywords that belong to non_key detected in three tokens before the gaz_current. For the definition of token, please refer to the section 2.1.1 about pre-processing.

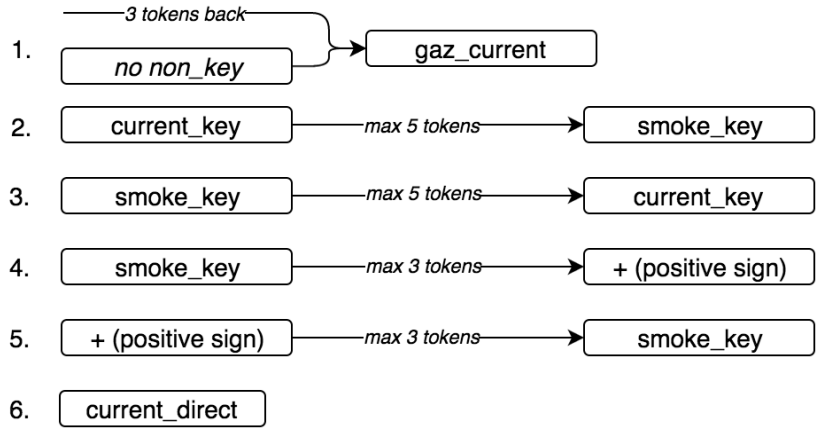

Figure 5.5. The JAPE rules designed to predict CURRENT SMOKER

Examples of each rule in Figure 5.5:

1. Rule number 1
    - *He is a **current smoker*** [VALID]
    - *He is **not** a **current smoker*** [INVALID] – there is a "not" in the third token behind "smoker".
2. Rule number 2
    - ***Continue** to **smoke** until now* [VALID]
    - *The patient is a **current** heavy alcohol drinker, but he does not **smoke*** [INVALID] – the distance between "currently" and "smoke" is more than 8 tokens.
3. Rule number 3
    - ***Smokes currently*** [VALID]
4. Rule number 4
    - ***Tobacco:** + for 40 years* [VALID]
5. Rule number 5
    - **+ *smoke*** [VALID]
6. Rule number 6
    - *He **smokes** one pack per day* [VALID]

5.4.2. Past Smoker Rules

Four rules to detect PAST SMOKER have been developed. The rules are more complex compared to the rules to detect CURRENT SMOKER. Even though the PAST SMOKER status is expressed in various ways in the training data, there are general patterns which are not specific to only one or two records. The rules are illustrated in Figure 5.6.

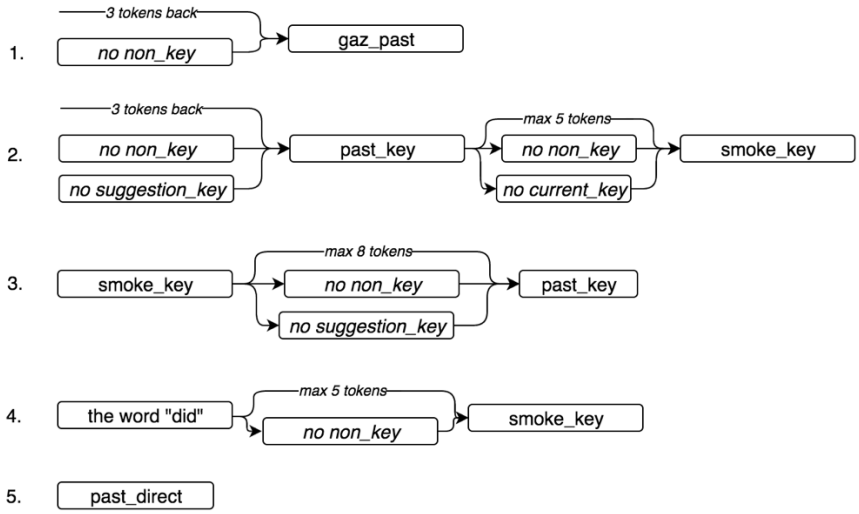

Figure 5.6. The JAPE rules designed to predict PAST SMOKER

Examples of each rule in Figure 5.6:

1. Rule number 1
    - *The patient is a **past smoker*** [VALID]
2. Rule number 2
    - *He **smoked** in the **past*** [VALID]
    - *It is **important** to **stop** being a **smoker*** [INVALID] – there is a suggestion word "important" behind "stop".
3. Rule number 3
    - ***Tob**: active for several years, **quit** one year ago* [VALID]
    - ***Smoke**: **trying** to **stop*** [INVALID] – there is a word "trying" between "smoke" and "stop" which is grouped as suggestion_key.
4. Rule number 4
    - ***Did smoke** in the past* [VALID]
    - ***Did not smoke** in the past* [INVALID] – there is a non_key between "did" and "smoke".
5. Rule number 5
    - *He **smoked** and drank heavily when he was child* [VALID] – the word "smoked" is considered as an occurrence in the past.

5.4.3. Non-smoker Rules

NON-SMOKER is the status with the fewest mention variations. The generated rules are also generally simpler than PAST SMOKER. The illustration of these rules can be seen in Figure 5.7.

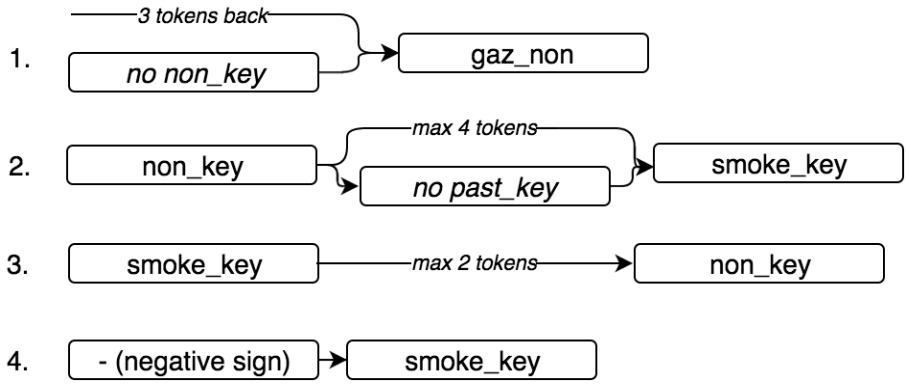

Figure 5.7. The JAPE rules designed to predict NON-SMOKER

Examples of each rule in Figure 5.7:

1. Rule number 1
    - *The patient is a* ***non-smoker*** [VALID]
    - *He* ***denied*** *being a* ***non smoker*** [INVALID] – a double negations detected in the sentence, as "denied" is grouped as non_key.
2. Rule number 2
    - *He never smoked* [VALID]
    - *He does* ***not discontinue smoking*** [INVALID] – there is a past_key "discontinue" in between "not" and "smoking".
3. Rule number 3
    - ***Tob****: neg* [VALID]
    - ***Smoke****: positive, alcohol:* ***negative*** [INVALID] – the distance between "smoke" and "negative" is more than 2 tokens.
4. Rule number 4
    - ***-tobacco*** [VALID]
    - *2-5 packs of* ***cigarette*** *a week* [INVALID] – the distance between (-) and the word "cigarette" is too far.

5.4.4. Rules Index Tables

Below are the index tables for the keywords used in the rules creation. Three types of keywords are defined in the "Type" column:

- Substring: The rules will fire if the defined substrings are detected in a word (i.e. *smok* is a substring of *smoke*, *smoking*, or *smokes*). The example of the complete words from a substring can be seen in the "Example" column.
- Exact match: The word should be exactly the same (but case insensitive) to the lists to make the rules fire.
- Direct: Depending on the category, if the words in this type found in text, the rules will fire (e.g. if the word "*ex-tob*" found in a text, a phrase-level prediction of PAST SMOKER will be given).

Table 5.2. Smoking keywords index

Smoking keywords	Type	Example	Group
smok cigar tobac	substring	smoke, smoking, smoked cigar, cigarette tobac, tobacco	smoke_key
nicotine, tob, cig, cigs	exact match	-	

Table 5.3. Current smoker keywords index

Current smoker keywords	Type	Example	Group
continu	substring	continue, continuing	current_key
has, still, + (positive sign)	exact match	-	
smokes, smoker, smoking	direct	-	current_direct

Table 5.4. Past smoker keywords index

Past smoker keywords	Type	Example	Group
please, important, must, should, trying, try, if	exact match	-	suggestion_key
stop discontinu quit	substring	stop, stopped, stopping discontinue, discontinued, discontinuing quit, quitted	past_key
remote, past, former, prior, ex-, quit	exact match	-	
smoked, ex-tob	direct	-	past_direct

Table 5.5. Non-smoker keywords index

Non-smoker keywords	Type	Example	Group
neg	substring	neg, negative	non_key
no, non, none, not, nor, deny, denies, denying, never, abstain, n't, - (negative sign)	exact match	Note: "n't" as in "don't", because the tokeniser divide "don't" into "do" and "n't"	

Below are the gazetteer entries defined in the system. The gazetteer is divided into four lists:

- current_smoker.lst: contains obvious mentions of CURRENT SMOKER
- non_smoker.lst: contains obvious mentions of NON-SMOKER
- past_smoker.lst: contains obvious mentions of PAST SMOKER

- family.lst: contains a list of family member names. If a sentence contains a word on this list, it will be excluded from the detection to prevent false matching (see section 3.3.3.3 about error prevention).

Table 5.6. Index of gazetteer entries

List name	Entries	Group
current_smoker.lst	currently smoking current smoker	gaz_current
non_smoker.lst	nonsmoker non smoker non-smoker nonsmoking non smoking non-smoking	gaz_non
past_smoker.lst	exsmoker ex smoker ex-smoker former smoker past smoker prior smoker remote smoker	gaz_past
family.lst	aunt, aunts, brother, brothers, child, children, cousin, cousins, daughter, daughters, father, grandfather, grandmother, mother, nephew, nephews, niece, nieces, parent, parents, sister, sisters, son, sons, uncle, uncles	-

5.5. Smoking Status Precedence

The next step is to decide the smoking status precedence if more than one phrase-level predictions are found in a record. The three smoking categories (CURRENT SMOKER, PAST SMOKER, and NON-SMOKER) have been examined. There is no

need to examine the other two categories (SMOKER and UNKNOWN), as the priority for them has already been decided: SMOKER will only be given if smoking mentions are in the record but evidence is insufficient to classify it further, while UNKNOWN will be given if no smoking mention is found in the record.

The precedence has been decided by measuring the "likeliness" of a smoking status to be correctly predicted. A matrix that contains the comparison between the total records of each smoking category in a dataset and the total phrase-level predictions detected in each category has been created for each dataset, and can be seen in Table 5.7 for the i2b2 2006 data and Table 5.8 for the i2b2 2014 data.

The numbers in both tables show the total phrase-level predictions detected for each category of gold standard and prediction. The numbers inside the parentheses show the relative score of the number in that cell with the total number of records in a specific label.

Both tables show that NON-SMOKER is the least ambiguous prediction. This means that if a record contains a phrase-level prediction of NON-SMOKER, only a small chance exists that it contains phrase-level prediction of other smoking categories. Hence, the NON-SMOKER precedence can be safely placed after the two other smoking categories.

Table 5.7. The comparison matrix for the i2b2 2006 training set

Gold standard labels	Total records	Total phrase-level prediction detected				
		CURRENT SMOKER	PAST SMOKER	NON-SMOKER	SMOKER	UNKNOWN
CURRENT SMOKER	35	**33 (0.94)**	2 (0.06)	0 (0.00)	0 (0.00)	1 (0.03)
PAST SMOKER	36	8 (0.22)	**34 (0.94)**	1 (0.03)	2 (0.05)	0 (0.00)
NON-SMOKER	66	0 (0.00)	0 (0.00)	**64 (0.98)**	2 (0.03)	0 (0.00)
SMOKER	9	4 (0.44)	1 (0.11)	0 (0.00)	3 (0.33)	0 (0.00)
UNKNOWN	252	0 (0.00)	0 (0.00)	1 (0.00)	1 (0.00)	250 (0.99)

Table 5.8. The comparison matrix for the i2b2 2014 training set

Gold standard	Total records	Total phrase-level prediction detected				
		CURRENT SMOKER	PAST SMOKER	NON-SMOKER	SMOKER	UNKNOWN
CURRENT SMOKER	58	**49 (0.84)**	3 (0.05)	7 (0.12)	5 (0.09)	1 (0.02)
PAST SMOKER	149	48 (0.32)	**130 (0.87)**	30 (0.20)	4 (0.03)	4 (0.03)
NON-SMOKER	184	4 (0.02)	4 (0.02)	**178 (0.97)**	2 (0.01)	2 (0.01)
SMOKER	9	5 (0.55)	3 (0.33)	1 (0.11)	0 (0.00)	1 (0.11)
UNKNOWN	371	1 (0.00)	1 (0.00)	4 (0.01)	2 (0.00)	364 (0.98)

The most ambiguous prediction is PAST SMOKER. As an example, Table 5.8 shows that the proportion of the records labelled with PAST SMOKER that contain phrase-level predictions of PAST SMOKER is 0.87, a significantly high proportion. However, the records also contain a large proportion of phrase-level predictions for CURRENT SMOKER (0.32) and NON-SMOKER (0.20). On the other side, from Table 5.8 it can be inferred that if a record contains a phrase-level prediction of PAST SMOKER, most likely it is the correct prediction for that record. Only three records (0.05) that were belong to CURRENT SMOKER and four records (0.02) that were belong to NON-SMOKER, compared to the 130 correctly classified records (0.87) as PAST SMOKER. Therefore, to avoid misclassification, the PAST SMOKER should be placed as the first precedence.

The CURRENT SMOKER label has a likeliness level between PAST SMOKER and NON-SMOKER, according to the table. Thus, it can be placed in the middle. The hierarchy for the record-level prediction can be seen in Figure 5.8.

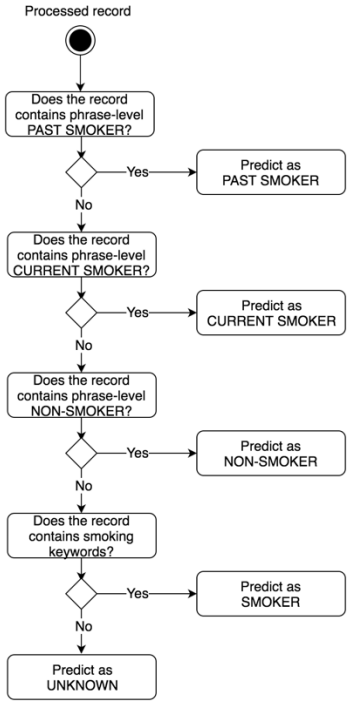

Figure 5.8. Smoking status precedence hierarchy

5.6. Alcohol Consumption Status Prediction

The alcohol consumption status extractor has been developed based on the smoking status extractor. The keywords related to smoking condition were replaced with the corresponding keywords related to alcohol consumption. The process was done manually by examining the records in the i2b2 2006 and i2b2 2014 datasets. Both train datasets contain no alcohol consumption annotation. Hence, the classification judgment of a record was based on a non-expert annotator: the researcher of this project.

Table 5.9. shows the conversion of smoking categories to alcohol consumption categories. The five alcohol consumption categories have a comparable definition to the smoking categories, which can be reviewed in section 3.2.2 regarding data specification. No gazetteer entry developed for alcohol consumption predictor, as the terms such as *currently drinking*, *non-drinker*, or *past drinker* are not common in this case. Table 5.10 shows the conversions between smoking and alcohol consumption terms.

Table 5.9. The equivalent categories for alcohol consumption

Smoking status category	Alcohol consumption equivalent
CURRENT SMOKER	CURRENT DRINKER
PAST SMOKER	PAST DRINKER
NON-SMOKER	NON-DRINKER
SMOKER	DRINKER
UNKNOWN	UNKNOWN

Table 5.10. Alcohol consumption keywords

Smoking status	Alcohol consumption status
Keywords	
smok, cigar, tobac, nicotine, tob, cig, cigs	drink, drank, alcohol, etoh, ethanol
To predict current conditions	
smokes, smoker, smoking	drinks, drinker, drinking, alcohol, etoh, ethanol
To predict past conditions	
smoked, ex-smoker, exsmoker	drank, ex-drinker, exdrinker
To predict negative conditions	
nonsmoker, non-smoker, nonsmoking, non-smoking	nondrinker, non-drinker, nondrinking, non-drinking

The development performance of the alcohol consumption status extractor cannot be evaluated because no annotated training data exists for alcohol consumption status. However, the system has been tested with a dataset from The Christie's hospital, which will be discussed in chapter 6.

5.7. Testing

A series of tests have been done at the end of each development iteration to make sure that the system works as intended. The tests were divided into five categories based on the phases in the project: data preparation, pre-processing, information extraction, post-processing, and evaluation. The summary of the tests is presented in Table 5.11.

Table 5.11. System evaluation summary

Stage	Tests	Status
Data preparation	The system excludes the records without smoking status annotation	Passed
	The system combines the data into a correct XML format	Passed
Pre-processing	The system generates appropriate tokens in the records	Passed
	The system recognise the boundary of the sentences and not overflow the prediction annotations across sentences	Passed
Information extraction	The gazetteer entries work with the supplied text	Passed
	The sentences that marked as PAST SMOKER or NON-SMOKER do not have CURRENT SMOKER annotations.	Passed
	The rules work as intended (The cases have been checked by examining particular parts in the records or creating artificial records to test out the rules durability)	Passed
Post-processing	The record-level classification works based on the smoking status precedency level	Passed
Evaluation	The system calculates the evaluation measurements (precision, recall, F-measure) accurately	Passed

5.8. Development Results

The detailed result of each development iteration using the i2b2 2006 and i2b2 2014 training data can be seen in Table 5.12. The term *P* denotes precision, *R* stands for recall, and *F* indicates F-measure.

Table 5.12. Development evaluation result with training data

Iteration	CURRENT SMOKER			PAST SMOKER			NON-SMOKER			SMOKER			UNKNOWN		
	P	R	F	P	R	F	P	R	F	P	R	F	P	R	F
i2b2 2006 data															
Iteration 1	1.00	0.43	0.60	0.76	0.94	0.84	0.95	0.95	0.95	0.42	0.89	0.57	0.99	0.99	0.99
Iteration 2	1.00	0.51	0.68	0.85	0.92	0.88	0.98	0.97	0.98	0.32	0.89	0.47	1.00	0.99	0.99
Iteration 3	0.85	0.83	0.84	0.85	0.94	0.89	0.98	0.97	0.98	0.38	0.33	0.35	1.00	0.99	0.99
Iteration 4	0.86	0.91	0.89	0.92	0.94	0.93	0.98	0.97	0.98	0.38	0.33	0.35	1.00	0.99	0.99
i2b2 2014 data															
Iteration 1	0.62	0.28	0.38	0.74	0.79	0.76	0.83	0.66	0.74	0.08	0.33	0.12	0.91	0.98	0.95
Iteration 2	0.77	0.47	0.58	0.89	0.84	0.87	0.92	0.93	0.93	0.10	0.44	0.17	0.98	0.98	0.98
Iteration 3	0.70	0.72	0.71	0.89	0.84	0.87	0.94	0.92	0.93	0.00	0.00	0.00	0.98	0.98	0.98
Iteration 4	0.76	0.83	0.79	0.92	0.87	0.89	0.94	0.93	0.94	0.00	0.00	0.00	0.98	0.98	0.98

The summary of microaveraged and macroaveraged F-measure are shown in Table 5.13.

Table 5.13. Microaveraged and macroaveraged F-measures for the train datasets

Iteration	i2b2 2006 data		i2b2 2014 data	
	Macroaveraged F-measure	Microaveraged F-measure	Macroaveraged F-measure	Microaveraged F-measure
Iteration 1	0.79	0.93	0.59	0.81
Iteration 2	0.80	0.94	0.71	0.90
Iteration 3	0.81	0.95	0.70	0.91
Iteration 4 (final)	0.83	0.96	0.72	0.93

The graphs which summed the F-measure scores in each iteration for each smoking status can be seen in Figures 5.9 and 5.10.

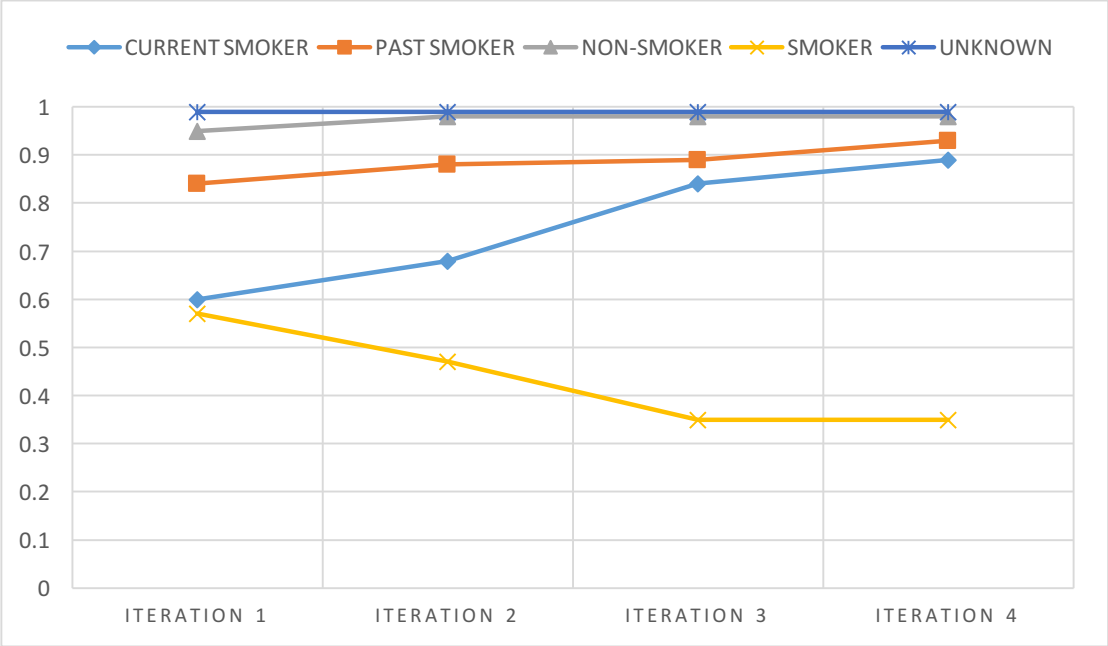

Figure 5.9. F-measure of each smoking status in the i2b2 2006 dataset

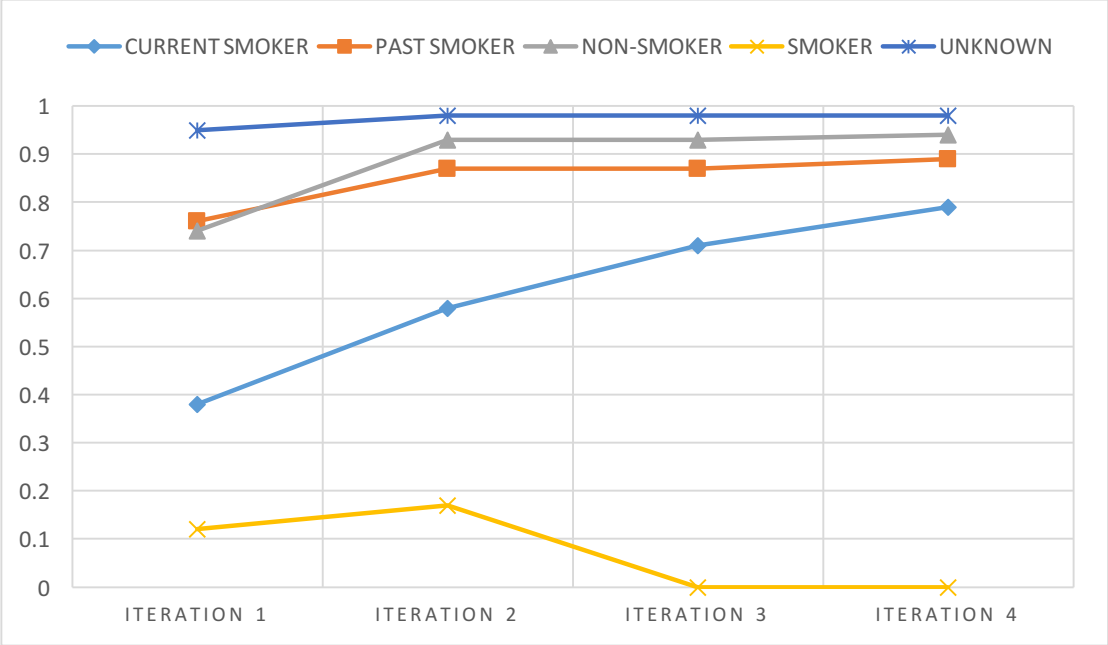

Figure 5.10. F-measure of each smoking status in the i2b2 2014 dataset

The graphs show the change of the F-measure through the four iterations. It can be inferred that UNKNOWN is the easiest category to predict as it has the highest F-measure scores in both datasets. NON-SMOKER, which has many straightforward

mentions, also gains a high accuracy on both datasets, and its scores are improved along the iterations. PAST SMOKER gains a slightly lower score compared to NON-SMOKER.

The two problematic categories are CURRENT SMOKER and SMOKER. There are various ways to mention the state of CURRENT SMOKER, which cannot be generalised by rules. The physicians often describe CURRENT SMOKER with unobvious terms without time mentions (e.g. *smokes 1ppd*, *condition: smoking*, etc.). From the 2nd iteration, a new decision was made to include the smoke mentions that indicate present condition to CURRENT SMOKER even if there are no time-specific mentions. It increases the score for CURRENT SMOKER for the further iterations, but at the same time it decreases the scores for SMOKER. The decision will result in higher microaveraged F-measure scores but lower macroaveraged F-measures, as the F-measure of SMOKING status has been pushed down.

This decision was made because only 1% to 2% of records contain SMOKING status in either dataset, which is not enough to make robust rules. This condition is understandable since the SMOKING status was only given if the annotators could not decide on a more specific status for the record. That condition explains the nature of the ambiguity in SMOKING status.

Figure 5.11 shows the changes in the microaveraged and macroaveraged scores for both train datasets in each iteration. In general, the scores are increasing along the iterations. The scores for the i2b2 2014 data are increased significantly from iteration 1 to iteration 2. This is because the i2b2 2014 data was not used in the first iteration. Thus, there are a number of undetected mentions in the data.

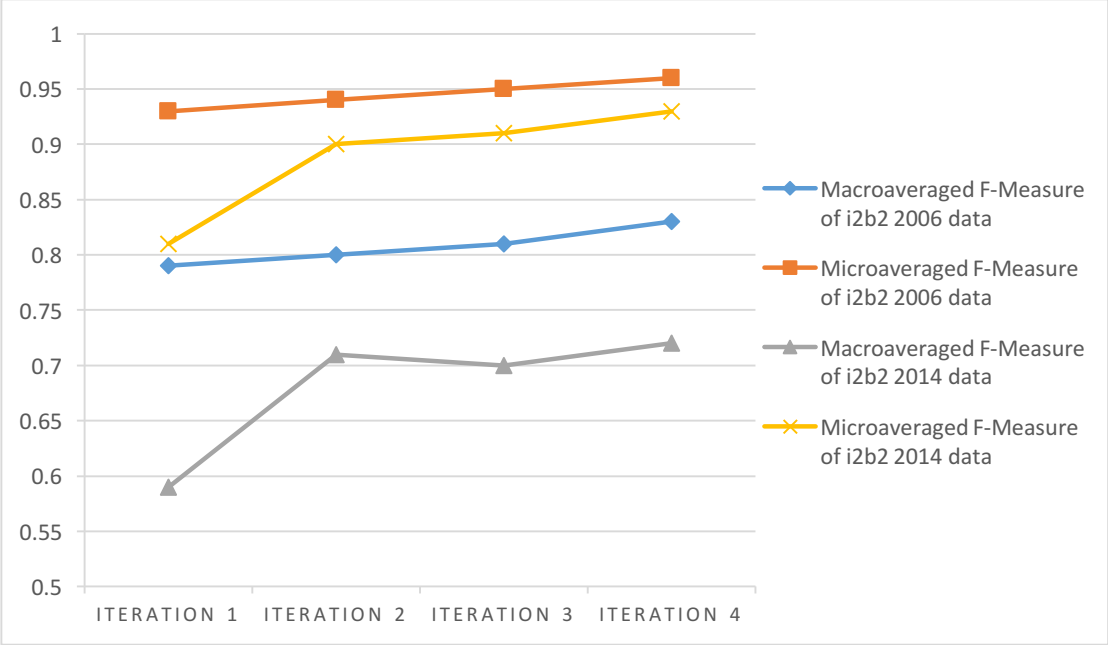

Figure 5.11. Changes in the macroaveraged and microaveraged F-measures for the i2b2 2006 and the i2b2 2014 train datasets

6. Results and Discussion

This chapter discusses the results of applying testing data to the developed system. The results were analysed and the errors were identified for further development. Furthermore, a discussion about the inconsistencies in the datasets is also included in the chapter.

6.1. Results

This section discusses the performance of the system by using the datasets which were not used in the development process. In addition to the i2b2 2006 and i2b2 2014 training sets, an anonymised dataset from The Christie NHS Foundation Trust in Manchester was used. The Christie's data was originally unseen data from a different source than the other datasets. It was meant to test out the system in a real-world situation.

Table 6.1 shows an overview of the testing sets that were used to evaluate the system. Due to the limitation of the resources, only the data from The Christie was used to evaluate the alcohol consumption predictor. The smoking and alcohol consumption status gold standards in the Christie's data were annotated by a non-expert annotator. The detailed statistics of all training sets can be seen in Table 6.2 for smoking status and Table 6.3 for alcohol consumption status.

Table 6.1. Description of the test datasets

Testing set	Used to evaluate	Annotation	Total Records
i2b2 2006	smoking	Annotated by two expert annotators (already done by the i2b2 team)	104
i2b2 2014	smoking		502
The Christie	smoking, alcohol consumption	Annotated by a non-expert annotator, but the confusing cases were checked by a clinical informatician	88 (for smoking extraction), 89 (for alcohol consumption extraction)

Table 6.2. Statistics of the smoking status labels in the test datasets

Smoking Categories	i2b2 2006 test set	i2b2 2014 test set	The Christie test set
CURRENT SMOKER	11	33	4
PAST SMOKER	11	113	14
NON-SMOKER	16	120	28
SMOKER	3	3	1
UNKNOWN	63	243	41
Total records	***104***	***512***	***88***

Table 6.3. Statistics of the alcohol consumption status labels in the test datasets

Alcohol Consumption Categories	The Christie test set
CURRENT DRINKER	24
PAST DRINKER	1
NON-DRINKER	9
DRINKER	0
UNKNOWN	55
Total records	***89***

6.1.1. The i2b2 2006 Testing Set

The evaluation of the system using the i2b2 2006 testing set can be seen in Table 6.4. The results of this project are comparable to the 1st and 2nd ranked participants of the i2b2 2006 challenge (see section 2.3.4 which discusses the i2b2 2006 challenge result). The SMOKER, which received a zero F-measure score, is the most problematic label in the dataset. However, only 3 records were annotated as SMOKER in the dataset. CURRENT SMOKER also received a smaller score compared to PAST SMOKER and NON-SMOKER, while UNKNOWN achieved the

highest score. This is consistent with the analysis of the training set (see section 5.8 about development results).

Table 6.4. i2b2 2006 testing set scores

Smoking categories (total)	Measurement		
	Precision	**Recall**	**F-measure**
CURRENT SMOKER (11)	0.58	0.64	0.61
PAST SMOKER (11)	0.82	0.82	0.82
NON-SMOKER (16)	0.93	0.88	0.90
SMOKER (3)	0.00	0.00	0.00
UNKNOWN (63)	0.98	1.00	0.99
Macroaveraged F-measure	**0.66**		
Microaveraged F-measure	**0.89**		

6.1.2. The i2b2 2014 Testing Set

The evaluation using the i2b2 2014 testing set shows a similar result to the i2b2 2006 data. The UNKNOWN category received the highest score, while CURRENT SMOKER and SMOKER received lower scores. The overall microaveraged and macroaveraged F-measure scores are slightly higher than the i2b2 2006 scores.

Table 6.5. i2b2 2014 testing set scores

Smoking categories (total)	Measurement		
	Precision	**Recall**	**F-measure**
CURRENT SMOKER (33)	0.57	0.70	0.63
PAST SMOKER (113)	0.89	0.85	0.87
NON-SMOKER (120)	0.94	0.84	0.89
SMOKER (3)	0.06	0.33	0.11
UNKNOWN (243)	0.99	0.98	0.98

Macroaveraged F-measure	**0.70**
Microaveraged F-measure	**0.91**

6.1.3. The Christie NHS Foundation Trust Data

An unannotated and anonymised dataset was randomly retrieved from the repository of The Christie NHS Foundation Trust in Manchester, UK. It contains 89 records (one record for smoking status prediction was missing) of clinical narratives of patients that suffered various types of cancer.

The dataset was converted to a suitable XML format and was passed through the system. The results for the smoking status prediction of this dataset can be seen in Table 6.6. The score is quite similar to the other datasets, where UNKNOWN, NON-SMOKER, and PAST SMOKER gained high scores, while CURRENT SMOKER and SMOKER received lower scores. However, only four records of CURRENT SMOKER and one record of SMOKER were in the dataset, which is insufficient to make a general performance conclusion for both categories.

Table 6.6. The Christie's data testing set scores for smoking prediction

Smoking categories (total)	**Measurement**		
	Precision	**Recall**	**F-measure**
CURRENT SMOKER (4)	0.76	0.50	0.57
PAST SMOKER (14)	0.93	0.93	0.93
NON-SMOKER (28)	1.00	1.00	1.00
SMOKER (1)	0.00	0.00	0.00
UNKNOWN (41)	0.98	1.00	0.99
Macroaveraged F-measure	**0.70**		
Microaveraged F-measure	**0.95**		

The modified system to predict alcohol consumption status obtained similar results compared to the smoking status predictor. UNKNOWN and NON-DRINKER were

among the ones which got high scores. Unlike the CURRENT SMOKER, CURRENT DRINKER gained a good performance with a perfect precision score. The zero scores of PAST DRINKER cannot be generalised, as there is only one record in the dataset which was marked with this category. The DRINKER category was excluded from the calculation since there is no single record marked with this label.

Table 6.7. The Christie's data testing set scores for alcohol consumption prediction

Alcohol Consumption Categories (total)	**Measurement**		
	Precision	**Recall**	**F-measure**
CURRENT DRINKER (24)	1.00	0.88	0.93
PAST DRINKER (1)	0.00	0.00	0.00
NON-DRINKER (9)	0.82	1.00	0.90
DRINKER (0)	*inapplicable*		
UNKNOWN (55)	1.00	1.00	1.00
Macroaveraged F-measure	**0.71**		
Microaveraged F-measure	**0.96**		

From the results discussed above, it can be inferred that the difference between the scores in the development stage and in the testing stage were comparable with only a slight drop in the macroaveraged F-measure scores. Thus, it can be concluded that the system is neither over-fit nor under-fit to the training data.

6.2. Discussion

This section discusses the result of the system and analyses the classification errors that occurred in the testing set. The analysis was done by comparing the system's output with the gold standards and manually inspecting the records that were falsely classified.

6.2.1. Error Analysis

The error analysis was done by tabulating the output from the system in a spreadsheet and manually inspecting the errors, as can be seen in Figure 6.1. The errors were discussed in accordance with their gold standard labels.

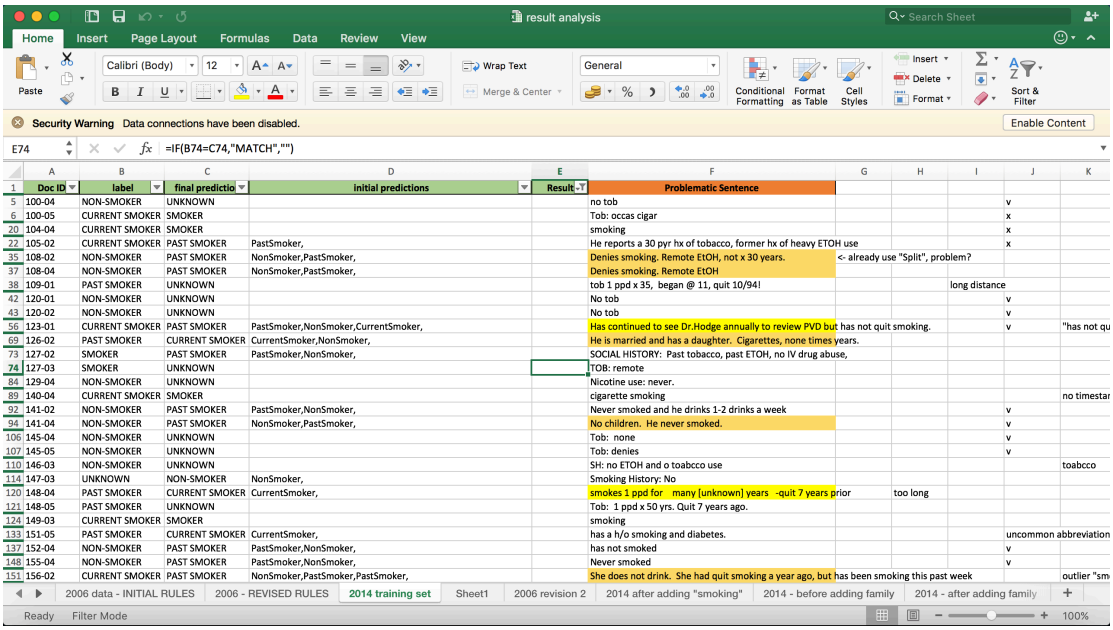

Figure 6.1. Analysing the errors by using spreadsheets

1. Errors in CURRENT SMOKER

   Most of the errors in the records that were labelled with CURRENT SMOKER were because of the failure in recognising the temporal expressions around the smoking keywords. Eight of ten records that should be classified as CURRENT SMOKER were predicted as PAST SMOKER due the temporal ambiguities. Some examples of the entries on the misclassified records are shown in Table 6.8.

Table 6.8. Examples of CURRENT SMOKER records that were incorrectly classified

Entry	Prediction	Description
quit *smoking in June, former 1 ppd x 60 years.*	PAST SMOKER	Quit less than one year
Smoking: 1 pack/day x 40 years. He ***stopped*** *one year ago and now smokes intermittently 2-3 cigarettes/day.*	PAST SMOKER	Quit smoking but repeated his habit recently

*...despite being told she had to **quit** smoking.*	PAST SMOKER	Only a suggestion
***Interesetd** on **quiting** smoking.*	PAST SMOKER	Typo = interesetd, the gazetteer failed to recognise the suggestion_key
***Former** heavy smoker, she has been unable to smoke as of three months ago*	PAST SMOKER	Quit less than one year

All of the examples were predicted as PAST SMOKER since there are keywords that indicate past conditions. However, the real condition is quite different. Either the patients stopped smoking less than one year ago (indicated by the temporal mentions), had stopped smoking in the past but were practicing the habit currently, or just showed an expression of interest to quit smoking. Typos in the sentence can also cause an error, as can be seen in the fourth example.

2. Errors in PAST SMOKER

   The errors in PAST SMOKER were due to failures to recognise the main context in the sentence, as the condition of patients that smoked in the past is often not written in a direct manner. Another reason is that some smoking conditions are mentioned in more than one sentence, and each sentence supports the other (e.g. *the patient has a smoking history. He then quit in 2014*). The system failed to recognise the separated key smoking context in the second sentence. Most of erroneous PAST SMOKER records were falsely predicted as CURRENT SMOKER or NON-SMOKER. The examples are shown in Table 6.9.

Table 6.9. Examples of PAST SMOKER records that were incorrectly classified

Entry	Prediction	Description
*60 pk year smoking history, but **none** currently.*	NON-SMOKER	Failed to understand the whole context of the sentence
*Tobacco: **none** for years*	NON-SMOKER	Failed to understand the whole context of the sentence
*still **not** smoking after 13 months!*	NON-SMOKER	Failed to understand the whole context of the sentence
*He does **not** currently smoke or drink alcohol.*	NON-SMOKER	Failed to understand the whole context of the sentence
***Smoking** hx: 1/2-1ppd x 30 yrs, quit 2153*	CURRENT SMOKER	The distance between smoking keyword and the word "quit" is too far
*approximately 100 pack/year **smoking** history. Quit.*	CURRENT SMOKER	The keyword "quit" is in a different sentence.

Another aspect that made the identification more difficult was the data anonymisation. It will alter any temporal mentions, and those mentions are generally useful to predict the current smoking condition of a patient (see the fifth example).

3. Errors in NON-SMOKER

   NON-SMOKER is one of the labels that has been predicted accurately. The errors were mainly because the distance between the smoking keywords and the keywords that indicate negative condition is out of range (see chapter 5.4 for the ranges of each rule). Another reason is because other sentences in the record contained smoking keywords and were predicted as CURRENT SMOKER or PAST SMOKER. As both categories have higher precedence than NON-SMOKER, the record will be falsely classified as one of them. The examples are shown in Table 6.10.

Table 6.10. Examples of NON-SMOKER records that were incorrectly classified

Entry	Prediction	Description
She does not drink , use IV drugs or smoke .	SMOKER	The distance between "does not" and the word "smoke" is too far
***formerly** worked in mathmatics- does not smoke,*	PAST SMOKER	Misclassified by the word "formerly"
*Denies tobacco use in **past** or **currently**.*	PAST SMOKER	Ambiguous words "in past", thus contains a phrase-level PAST SMOKER, which has higher precedence than NON-SMOKER
There is no history of drug abuse or smoking .	CURRENT SMOKER	The distance between "no" and the word "smoking" is too far

4. Errors in SMOKER

   SMOKER is the most ambiguous and problematic category. Only a very small number of records were marked with this category in both training and test datasets. The annotators only gave this label to the records when, based on their knowledge, they were unsure of the current condition of the patient. Consequently, since the rules were made based on the similar human knowledge, it is quite challenging to handle the ambiguity. This issue was also discussed in the i2b2 2006 and i2b2 2014 summary paper [6] [23]. This category also received low precisions in both training and testing data, since a number of records could not be identified to either CURRENT SMOKER, PAST SMOKER, or NON-SMOKER by the system due to the absence of the supporting keywords, and then they were falsely predicted as SMOKER. The examples are shown in Table 6.11.

Table 6.11. Examples of SMOKER records that were incorrectly classified

Entry	Prediction	Description
Smoking: 2ppd x 54 yrs	CURRENT SMOKER	Contains the word "smoking", unclear condition
A 30-pack-year cigarette smoking history and 6 drinks per day .	CURRENT SMOKER	Contains the word "smoking", unclear condition
+smoking	CURRENT SMOKER	Contains the word "smoking"
The patient has history of heavy cigar smoking , no cigarettes .	CURRENT SMOKER	Unclear condition "heavy cigar smoking, no cigarettes"

Some of the similar mentions to those above were classified as other smoking status. The inconsistency of the gold standards will be discussed in section 6.2.2.

5. Errors in UNKNOWN

   UNKNOWN is the category with the highest score. This status was given to the records if there was no smoking keyword detected within them, or the annotators disagreed about the smoking condition of the patient. The errors occurred if smoking related keywords were in the record, but they were not about the patient's condition. The examples are shown in Table 6.12.

Table 6.12. Examples of UNKNOWN records that were incorrectly classified

Entry	Prediction	Description
*Per prior notes, **no tobacco**, alcohol, illicits.*	NON-SMOKER	Ambiguous word "illicits"
smok cessation reinforced	SMOKER	Unclear sentence
Tob:unknown	SMOKER	Smoking keyword "tob" is mentioned but the status is unknown
Tobacco Fhx of CAD ETT 12/94 (-)	SMOKER	Unclear sentence

6. Errors in alcohol consumption prediction

   In general, the errors in alcohol consumption prediction were similar to the errors in the smoking categories since they were expressed in similar ways. However, there are some interesting aspects to note from the alcohol consumption prediction. The first aspect is that unlike the CURRENT SMOKER, the CURRENT DRINKER category got high scores. This is because the condition is normally mentioned in more explicit and obvious ways compared to the CURRENT SMOKER.

   Another aspect is that the PAST DRINKER condition is not commonly mentioned in the record. This is either because the physicians do not commonly take note of that condition for some reason, or it is unusual for alcohol drinkers to stop their habit.

   Unfortunately, due to privacy and confidentiality concerns, examples of the erroneous prediction from The Christie's data cannot be displayed in this report.

6.2.2. Gold Standard Inconsistencies

Despite the fact that the i2b2 datasets were annotated by two expert annotators, a number of inconsistencies were still found (see Table 6.13 for examples):

a) It was agreed that the task was to identify the explicitly mentioned smoking condition. However, some annotations were made based on implicit terms, which cannot be easily recognised by non-experts (example no. 1).
b) It also was agreed that PAST SMOKER will be given if the patients stopped smoking more than one year ago. However, a number of records were marked as PAST SMOKER but they had no temporal expressions that indicated the time when the patients stopped smoking (examples no. 3 and 6). This condition was also applied to CURRENT SMOKER. Some records were marked as CURRENT SMOKER even though there was no temporal expression that indicated the current condition of the patients (examples no. 1, 4, and 8).
c) Some arguable annotations were found in SMOKER and UNKNOWN categories. Some of the records that contained ambiguous smoking mentions

were classified as SMOKER, while others were marked as UNKNOWN (examples no. 7 and 9).

d) A number of records contained the current condition of a patient explicitly, but the labels did not match to that explicit expressions (examples no. 2 and 5).

Table 6.13. Examples of inconsistencies found in the datasets

No.	Sentence	Gold Standard	Expected Label
1	*1-2 packs per day .*	CURRENT SMOKER	UNKNOWN
2	*quit tobacco greater than 25 years ago*	CURRENT SMOKER	PAST SMOKER
3	*He is a heavy smoker and drinks 2-3 shots per day at times .*	PAST SMOKER	CURRENT SMOKER
4	*Tobacco use*	CURRENT SMOKER	SMOKER
5	*Cigs- no*	UNKNOWN	NON-SMOKER
6	*history of tobacco use*	PAST SMOKER	SMOKER
7	*disease, hypertension, smoking history*	SMOKER	SMOKER
8	*positive smoking history*	CURRENT SMOKER	SMOKER
9	*smok cessation reinforced*	UNKNOWN	SMOKER
10	*Started on transderm nitro. Totally must quit all cigars.*	UNKNOWN	CURRENT SMOKER

There are some possible causes of this condition. First, the annotators decided the gold standard for a record based on their justifications of the clinical information mentioned in text, which was not obvious by a non-expert. Second, human errors were occurred in the data annotation process. Despite all the inconsistencies, the gold standards were used in this research without any alteration, because they were made by experts.

7. Conclusion

This chapter concludes the work that was done in this project. It includes the summary of achievements, reflections, and future work to improve the performance and to extend the functionality of the system.

7.1. Summary

The aim of this project was to develop a system that could extract smoking and alcohol consumption status of patients from unstructured clinical narratives. A rule-based text mining system with five components (data preparation, pre-processing, information extraction, post-processing, and evaluation) was successfully developed. In addition, all of the system requirements that were labelled with "should have", "must have", and "could have" have been successfully fulfilled (see Table 3.2 about project requirements).

The system achieves microaveraged F-measure scores between 0.89 and 0.96 for various test datasets, including the datasets that was taken from The Christie NHS Foundation Trust in Manchester. It can be concluded that the system generalises well on totally different datasets, and results in the state-of-the-art microaveraged F-measure performances of more than 0.90 on average for both smoking and alcohol consumption prediction.

As a summary, here are the achievements that have been made throughout the project:

a) The background research about clinical text mining was done extensively to formulate a solid foundation for the project.
b) The previous attempts to extract social factors were studied and analysed to assess the strengths and weaknesses of each method.
c) The datasets from various sources have been analysed to gain insights about how social factors are expressed in clinical narratives.
d) A strong research methodology and system design have been formulated based on the knowledge that was attained from the previous achievement points (a, b, and c).
e) A rule-based text mining system to extract smoking status was implemented by using GATE framework and its plugins. The workflow was supported by

some Java-based scripts to handle the data preparation, pre-processing, and evaluation stages.

f) The developed system was repurposed to extract alcohol consumption status.

g) The system was evaluated by using datasets from various sources. This resulted in the state-of-the-art microaveraged F-measure performances of more than 0.90 on average for both smoking and alcohol consumption status extractions.

Alongside with the achievements above, here are the main contributions of this project:

a) The system with a state-of-the-art performance for smoking status extraction.

b) The repurposed system to extract alcohol consumption status with a comparable performance to the smoking status extractor.

c) The system worked well with datasets from various sources. This was proved by the performance of the system when it was tested with The Christie's dataset.

d) The system was published as an open source program on the GitLab repository of the University's School of Computer Science (https://gitlab.cs.man.ac.uk/groups/gnTEAM), and is in a process to be published on the GitHub repository of The Christie NHS Foundation Trust (https://github.com/christie-nhs-data-science), with the Apache 2.0 license (https://opensource.org/licenses/Apache-2.0).

7.2. Reflection

Extracting social factors from clinical narratives is a challenging task. The first reason is because the data is not freely available due to the confidentiality and privacy issues. In this project, an agreement with the i2b2 was made to obtain a limited amount of clinical narrative data. Apart from that, the evaluation process by using The Christie's data was done only on a designated machine with a limited time. These conditions are understandable since the data contained private information of those involved in medication processes. Thus, the available clinical records data should be used efficiently to develop the system.

The unstructured nature of clinical records was also a challenge. Social factors were expressed in many ways and in various parts on the clinical narratives by physicians. The developed system needed to recognise the possible forms of mention to extract social factors accurately.

The next reason is that the data was retrieved in an anonymised condition. This means that all of the possible sequences that may be used to identify the patients or the physicians have been altered, including dates and some temporal expressions. This condition made the current condition of patients (e.g. either they are CURRENT SMOKER or PAST SMOKER) harder to be identified.

There were also some challenges from the technical aspect. As an instance, an appropriate sentence splitting method was chosen to annotate the beginning and the end of each sentence, which is often not obvious from the machine side. Failure to annotate the sentences can result in degraded performance for information extraction. Typos and grammatically incorrect sentences, which are common in clinical narratives, also decreased the performance, as there were a number of errors that were caused by the inability of the developed system to recognise the typos.

As a summary of this reflection, various aspects should be considered to make a successful attempt in the clinical information extraction task. Aside from the conceptual and technical aspects, the legal aspect and confidentiality issues should also be considered to ensure that the data is handled appropriately.

The author gained countless experiences in this project, including:

a) Dealing with non-technical aspects when handling human-related data and information.
b) Handling clinical information extraction tasks and realising the future prospects in this field.
c) Writing a sound research report.
d) Learning the importance of interdisciplinary collaboration to advancing research and development in clinical text mining.

7.3. Future Work

Despite the high performance achieved by the developed system in this project, there are a number of recommended improvements to make the system more robust.

Improvements could also be made to extend the functionality of the system. Below are some recommendations for future work.

a) Implement a machine learning algorithm to support the rule-based system.
   The mention of social factor is sometimes not obvious. Thus, the proper rules cannot be crafted. To handle this condition, some additional tags can be inserted into tokens, such as Part-of-Speech tagger and Named Entity transducer. The ambiguous sequences can be passed into the machine learning algorithm along with their tags as features to be processed statistically. However, a big amount of annotated data is needed to make a good machine learning extractor.
b) Extend the text mining workflow by adding temporal expression detectors.
   It has been realised that a number of errors in the system were due because of the inability to detect temporal mentions (e.g. time, date, month, year) around the smoking/alcohol consumption mentions. Adding the temporal expressions detector to the information extraction phase is expected to increase the system performance.
c) Implement the pairwise comparison for the post-processing stage.
   An analysis to determine the precedence of smoking/alcohol consumption status has been proved to effectively detect the suitable label for a record. However, a small number of false matches occurred due to the final decision errors. A more advanced method, such as pairwise comparison, should be implemented to reduce the errors. This method works by assigning a specific label to the most frequent combination of phrase-level predictions on that category.
d) Identify the social factor events in addition to the status.
   Some records mention the periods or moments when the patient started or stopped smoking/consuming alcohol (e.g. *stopped since last 2 years, started to smoke since child, stopped consuming alcohol on this pregnancy*). This information could also be extracted as an addition to the status to provide additional information for the analyst or physician.
e) Extend the functionality of the system.
   This project proved that repurposing smoking status predictor to extract alcohol consumption status worked well. It can be extended to predict other

social factors, such as: drug abuse, medication, family, and pet histories. The results could be processed further, for example, to examine the relation between these social factors or to predict the quality of the patients' life based on their social factors. These developments will trigger further research in the healthcare institutions and will improve the medication experience of the patients.

7.4. Conclusion

A text mining system to extract smoking and alcohol consumption status from clinical narratives has been developed and it produces a state-of-the-art performance. The author learned that this project is just a beginning of the endless possibilities of further development in clinical text mining. A number of deeper studies should be done to make the aim of clinical text mining becomes a reality: to enhance the health quality of human beings.

Bibliography

[1] Massachusetts General Hospital. Clinical Recognition: Describing Practice Through Clinical Narratives. [Online]. http://www.mghpcs.org/ipc/Programs/Recognition/Describing.asp

[2] Azad Dehghan, "Mining Patient Journeys from Healthcare Narratives," The University of Manchester, PhD Thesis 2014.

[3] Deloitte, "Digital Health in the UK: An industry study for the Office of Life Sciences," 2015.

[4] Alasdair Liddell, Stephen Adshead, and Ellen Burgess, "Technology in the NHS," 2008.

[5] A National Center for Biomedical Computing. (2006) i2b2: Informatics for Integrating Biology & the Bedside. [Online]. https://www.i2b2.org/

[6] Ozlem Uzuner, Ira Goldstein, Yuan Luo, and Isaac Kohane, "Identifying Patient Smoking Status from Medical Discharge Records," *Journal of the American Medical Informatics Association*, 2008.

[7] Richard T. Herschel and Nory E. Jones, "Knowledge management and business intelligence: the importance of integration," *Journal of Knowledge Management*, 2005.

[8] Andreas Hotho, Andreas Nürnberger, and Gerhard Paaß, "A Brief Survey of Text Mining," *Ldv Forum*, vol. 20, May 2005.

[9] Gary Miner, *Practical text mining and statistical analysis for non-structured text data applications*. Oxford, United Kingdom: Elsevier, 2012.

[10] Ronen Feldman and James Sanger, *The Text Mining Handbook: Advanced Approaches in Analysing Unstructured Data*. Cambridge, UK: Cambridge University Press, 2007.

[11] Pasi Tapanainen Gregory Grefenstette, "What is a word, What is a sentence? Problems of Tokenization," 1994.

[12] Rebecca Dridan and Stephan Oepen, "Tokenization: Returning to a Long Solved Problem. A Survey, Contrastive Experiment, Recommendations, and Toolkit," in *50th Annual Meeting of the Association for Computational Linguistics*, vol. 2, Jeju, 2012.

[13] Jiří Maršík and Ondřej Bojar, "TrTok: A Fast and Trainable Tokenizer for Natural Languages," *The Prague Bulletin of Mathematical Linguistics*, October 2012.

[14] Moty Ben-Dov and Ronen Feldman, "Text Mining and Information Extraction," in *Data Mining and Knowledge Discovery Handbook*. US: Springer, 2010.

[15] The University of Sheffield. (2009) GATE JAPE Grammar Tutorial. [Online]. https://gate.ac.uk/sale/thakker-jape-tutorial/GATE%20JAPE%20manual.pdf

[16] Dhaval Thakker, Taha Osman, and Phil Lakin, "GATE JAPE Grammar Tutorial," The University of Sheffield, Tutorial 2009.

[17] Chung-Chi Huang and Zhiyong Lu, "Community challenges in biomedical text mining over 10 years: success, failure and the future," *Briefings in Bioinformatics*, 2016.

[18] Biocreative. (2016) Critical Assessment of Information Extraction in Biology. [Online]. http://www.biocreative.org/

[19] BioNLP Shared Task 2016. (2016) BioNLP-ST 2016. [Online]. http://2016.bionlp-st.org/

[20] Informatics for Integrating Biology and the Bedside. (2009) i2b2. [Online]. https://www.i2b2.org/NLP/Medication/

[21] Ozlem Uzuner, Imre Solti, and Eithon Cadag, "Extracting medication information from clinical text," *Journal of the American Medical Informatics Association (JAMIA)*, Sept-Oct 2010.

[22] A National Center for Biomedical Computing. (2014) i2b2 2014 Challenge. [Online]. https://www.i2b2.org/NLP/HeartDisease/

[23] Amber Stubbs, Christopher Kotfila, Hua Xu, and Özlem Uzuner, "Identifying risk factors for heart disease over time: Overview of 2014 i2b2/UTHealth shared task Track 2," *Journal of Biomedical Informatics*, 2015.

[24] Eiji Aramaki, Takeshi Imai, Kengo Miyo, and Kazuhiko Ohe, "Patient Status Classification by using Rule based Sentence Extraction and BM25-kNN based Classifier ," *i2b2 Workshop on Challenges in Natural Language Processing for Clinical Data*, 2006.

[25] Aaron M. Cohen, "Five-way Smoking Status Classification Using Text Hot-Spot Identification and Error-correcting Output Codes," *Journal of the American Medical Informatics Association (JAMIA)*, 2008.

[26] Cheryl Clark et al., "Identifying Smokers with a Medical Extraction System," *Journal of the American Medical Informatics Association (JAMIA)*, 2008.

[27] Xua Hu, Marianthi Markatou, Rositsa Dimova, Hongfang Liu, and Carol Friedman, "Machine learning and word sense disambiguation in the biomedical domain: design and evaluation issues," *BMC Bioinformatics 2006*, 2006.

[28] Padraig Cunningham and Sarah Jane Delany, "k-Nearest Neighbour Classifiers," University College Dublin, Dublin, 2007.

[29] Christopher J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, 1998.

[30] DSDM Consortium. (2014) The DSDM Agile Project Framework. [Online]. https://www.dsdm.org/content/moscow-prioritisation

[31] Amber Stubbs and Ozlem Uzuner, "Annotating risk factors for heart disease in clinical narratives for diabetic patients," *Journal of Biomedical Informatics*, vol. 58, May 2015.

[32] Azad Dehghan, Aleksandar Kovacevic, George Karystianis, John A. Keane, and Goran Nenadic, "Combining knowledge- and data-driven methods for de-identification of clinical narratives," *Journal of Biomedical Informatics*, vol. 58, July 2015.

[33] George Karystianis, Azad Dehghan, Aleksandar Kovacevic, John A. Keane, and Goran Nenadic, "Using local lexicalized rules to identify heart disease risk factors in clinical notes," *Journal of Biomedical Informatics*, vol. 58, June 2015.

[34] George Karystianis, Therese Sheppard, William G. Dixon, and Goran Nenadic1, "Modelling and extraction of variability in free-text medication prescriptions from an anonymised primary care electronic medical record research database," *BMC Medical Informatics and Decision Making*, vol. 16, 2016.

[35] The University of Sheffield. (2016) General Architecture for Text Engineering. [Online]. https://gate.ac.uk/sale/tao/splitch6.html#x9-1220006.2

[36] Informatics for Integrating Biology and the Bedside. (2016) Informatics for Integrating Biology and the Bedside. [Online]. https://www.i2b2.org/NLP/DataSets/Agreement.php

[37] The University of Sheffield. (2016) The General Architecture for Text Engineering. [Online]. https://gate.ac.uk/sale/tao/split.html

[38] The University of Sheffield. (2016) General Architecture for Text Engineering. [Online]. https://gate.ac.uk/sale/tao/splitch6.html#x9-1200006

[39] Frank Tsui and Orlando Karam, *Essentials of Software Engineering*, 2nd ed. Sudbury, USA: Jones and Bartlett Publishers.

[40] TutorialsPoint. (2016) Tutorials Point. [Online]. http://www.tutorialspoint.com/sdlc/sdlc_spiral_model.htm

[41] Aleksandra Ivaylova Nacheva, "Extracting And Analysing Information From Patient Records In Order To Track Disease Progression Over Time," The University of Manchester, Master's Dissertation 2015.

[42] Osediamen Osoba, "Information Extraction for Road Accident Data," The University of Manchester, Master's Dissertation 2015.

[43] Justin Zobel, *Writing for Computer Science*. London: Springer, 2014.

Appendix

The JAPE rules needed for smoking status prediction are attached below. The alcohol consumption prediction uses the same codes but different keywords (see section 5.6).

Appendix 1: JAPE GazetteerPredictor

```
Phase:gazetteerpredictor
Input: Token Sentence Lookup Split
Options: control=appelt

Rule: PastSmoker_Gazetteer
(
        ({!Token.string ==~ "(?i)no|non|none|not|denied"}
        {!Token.string ==~ "(?i)no|non|none|not|denied"}
        {!Token.string ==~ "(?i)no|non|none|not|denied"}
        {Lookup.majorType == gaz_past_smoker})
): predictor
-->
:predictor.GazetteerRules = {rule = PastSmoker_Gazetteer}

Rule: CurrentSmoker_Gazetteer
(
        ({!Token.string ==~ "(?i)no|non|none|not|denied"}
        {!Token.string ==~ "(?i)no|non|none|not|denied"}
        {Lookup.majorType == gaz_current_smoker})
): predictor
-->
:predictor.GazetteerRules = {rule = CurrentSmoker_Gazetteer}

Rule: NonSmoker_Gazetteer
(
        ({!Token.string ==~ "(?i)no|non|none|not|denied"}
        {!Token.string ==~ "(?i)no|non|none|not|denied"}
        {!Token.string ==~ "(?i)no|non|none|not|denied"}
        {Lookup.majorType == gaz_non_smoker})
): predictor
-->
:predictor.GazetteerRules = {rule = NonSmoker_Gazetteer}
```

Appendix 2: JAPE SmokeMention

```
Phase:smokemention
Input: Token Sentence Lookup
Options: control=appelt



/*
Rule: Eliminator
(
        {Sentence contains {Lookup.majorType == family_mention}} | {Sentence contains
{Token.string =~ "(?i)marijuana"}}


):elim
-->
:elim.eliminate = {}
*/



Rule: EliminatorMarijuana
(
        {Sentence contains {Token.string =~ "(?i)marijuana"}}


):elim
-->
:elim.eliminate = {}



Rule: SmokingWord
(
        ({Token.string =~ "(?i)smok|cigar|tobac|nicotine"})
        |
        ({Token.string ==~ "(?i)tob|cig"})
        |
        ({Token.string =~ "(?i)cigs"})
): predictor
-->
:predictor.SmokeMention = {rule = SmokingWord}
```

Appendix 3: JAPE PastSmoker

```
Phase:pastsmoker
Input: Token SmokeMention Split Sentence
Options: control=appelt


Rule: EliminatorMarijuana
(
        {Sentence contains {Token.string =~ "(?i)marijuana"}}


):elim
-->
:elim.eliminate = {}


Rule: PastSmoker
(
        ({!Token.string =~ "(?i)not|no|please|important|must|should|trying|try|if"}
        {!Token.string =~ "(?i)not|no|please|important|must|should|trying|try|if"}
        {!Token.string =~ "(?i)not|no|please|important|must|should|trying|try|if"}
        {Token.string =~ "(?i)stop|discontinu|quit|remote|past|former|ex-|prior",
!Token.string ==~ "(?i)never|non|not|no|quitting"}
        ({Token, !Split})[0,5]
        {SmokeMention.rule == SmokingWord})
        |
        ({SmokeMention.rule == SmokingWord}
        ({Token, !Split, !Token.string =~
"(?i)please|important|must|should|trying|try|if"})[0,8]
        {Token.string =~ "(?i)stop|discontinu|quit|remote|past|former|ex-|did",
!Token.string ==~ "(?i)never|non|not|no|quitting"})
        |
        ({!Token.string =~ "(?i)never|no"}
        {!Token.string =~ "(?i)never|no"}
        {!Token.string =~ "(?i)never|no"}
        {Token.string ==~ "(?i)smoked|ex-smoker|exsmoker|ex-tob"})
        |
        ({Token.string =~ "(?i)did"}
        {!Token.string ==~ "(?i)not"}
        ({Token, !Split})[0,5]
        {SmokeMention.rule == SmokingWord})
        |
        ({Token.string =~ "(?i)did"}
        {SmokeMention.rule == SmokingWord})
): predictor
-->
:predictor.SmokingIndicator = {rule = PastSmoker}
```

Appendix 4: JAPE NonSmoker

```
Phase:nonsmoker
Input: Token SmokeMention Split Sentence
Options: control=appelt

Rule: EliminatorMarijuana
(
        {Sentence contains {Token.string =~ "(?i)marijuana"}}

):elim
-->
:elim.eliminate = {}

Rule: NonSmoker
(
        ({Token.string ==~
"(?i)no|non|none|not|nor|deny|denies|denied|denying|never|negative|neg|abstain|n't"}
        ({Token, !Token.string =~ "(?i)stop|discontinu|quit", !Split})[0,4]
        {SmokeMention.rule == SmokingWord})
        |
        ({SmokeMention.rule == SmokingWord}
        ({Token, !Split})[0,2]
        {Token.string ==~
"(?i)no|non|not|none|nor|deny|denies|denied|denying|never|negative|neg|abstain"})
        |
        ({Token.string =~ "(?i)nonsmoker|non-smoker|nonsmoking|non-smoking"})
        |
        ({Token.string ==~ "(?i)-"}
        {SmokeMention.rule == SmokingWord})
): predictor
-->
:predictor.SmokingIndicator = {rule = NonSmoker}
```

Appendix 5: JAPE CurrentSmoker

```
Phase:currentsmoker
Input: Token SmokeMention Split SmokingIndicator Sentence
Options: control=appelt



Rule: EliminatorAnotherStatus
(
        {Sentence contains {SmokingIndicator.rule == PastSmoker}} | {Sentence contains
{SmokingIndicator.rule == NonSmoker}}

):elim
-->
:elim.eliminate = {rule = EliminatorAnotherStatus}

Rule: EliminatorMarijuana
(
        {Sentence contains {Token.string =~ "(?i)marijuana"}}

):elim
-->
:elim.eliminate = {}

Rule: CurrentSmoker
(
        ({Token.string =~ "(?i)has|still|continu|current|smokes|smoker"}
        ({Token, !Split})[0,5]
        {SmokeMention.rule == SmokingWord})
        |
        ({SmokeMention.rule == SmokingWord}
        ({Token, !Split})[0,5]
        {Token.string =~ "(?i)still|continu|current|smokes|smoker"})
        |
        ({Token.string ==~ "(?i)smokes|smoker|smoking"})
        |
        ({SmokeMention.rule == SmokingWord}
        ({Token, !Split})[0,3]
        {Token.string ==~ "\\+"})
        |
        ({Token.string ==~ "\\+"}
        ({Token, !Split})[0,3]
        {SmokeMention.rule == SmokingWord})
): predictor
-->
:predictor.SmokingIndicator = {rule = CurrentSmoker}
```

Appendix 6: JAPE SmokingPredictor

```
Phase:smokingpredictor
Input: RECORD SmokingIndicator SmokeMention GazetteerRules
Options: control=appelt


// Gazetteer rules

Rule: PastSmokerPredictor_Gazetteer
(
        {RECORD contains {GazetteerRules.rule == PastSmoker_Gazetteer}}
):predictor
-->
:predictor.Prediction = {PREDICTION = "PAST SMOKER"}

Rule: CurrentSmokerPredictor_Gazetteer
(
        {RECORD contains {GazetteerRules.rule == CurrentSmoker_Gazetteer}}
):predictor
-->
:predictor.Prediction = {PREDICTION = "CURRENT SMOKER"}

Rule: NonSmokerPredictor_Gazetteer
(
        {RECORD contains {GazetteerRules.rule == NonSmoker_Gazetteer}}
):predictor
-->
:predictor.Prediction = {PREDICTION = "NON-SMOKER"}

// Normal rules

Rule: PastSmokerPredictor
(
        {RECORD contains {SmokingIndicator.rule == PastSmoker}}
): predictor_pastsmoker
-->
:predictor_pastsmoker.Prediction = {PREDICTION = "PAST SMOKER"}


Rule: CurrentSmokerPredictor
(
        {RECORD contains {SmokingIndicator.rule == CurrentSmoker}}
): predictor_currentsmoker
-->
:predictor_currentsmoker.Prediction = {PREDICTION = "CURRENT SMOKER"}
```

```
Rule: NonSmokerPredictor
(
        {RECORD contains {SmokingIndicator.rule == NonSmoker}}
): predictor_nonsmoker
-->
:predictor_nonsmoker.Prediction = {PREDICTION = "NON-SMOKER"}

Rule: SmokerPredictor
(
        {RECORD contains SmokeMention}
): predictor_smoker
-->
:predictor_smoker.Prediction = {PREDICTION = "SMOKER"}


Rule: UnknownPredictor
(
        {!RECORD contains SmokeMention}
): predictor_unknown
-->
:predictor_unknown.Prediction = {PREDICTION = "UNKNOWN"}
```