MONASH University

Intelligent Audit Code Generation from Free Text in the Context of Neurosurgery

Sedigheh Khademi Habibabadi

Bachelor of Applied Mathematics, Isfahan University of Technology, Isfahan, Iran
Master of Socio-economical systems, Isfahan University of Technology, Isfahan, Iran
Master of Business Information Systems, Monash University, Melbourne, Australia

A thesis submitted for the degree of *Master of Philosophy* at
Monash University in 2016
Faculty of Information Technology

Copyright notice

Abstract

Clinical auditing requires structured data for aggregation and analysis of patterns. Clinicians however, need to record clinical encounters in written or spoken language, not only for its work-flow naturalness but also for its expressivity, precision, and capacity to convey all required information, which codified structured data is incapable of. Therefore, structured data must be obtained from clinical text as a later step, a task known as information extraction. Specialised areas of medicine use their own clinical language and clinical coding systems, resulting in unique challenges for the extraction process.

This research is devoted to creating a novel semi-automated method for generating codified auditing data from clinical notes recorded in a neurosurgical department in an Australian teaching hospital. The department has its own audit coding system, and language used in its clinical notes is highly specific to the neurological and neurosurgical domains, which necessitated a customised approach.

The principles of Design Science Research were followed to design a method that combines Natural Language Information Extraction and Machine Learning techniques. The method was tested by developing a computer programme that incorporates text extraction algorithms trained and tested on data supplied by the neurosurgical department. The software implements rules initially provided by a domain expert and extended during the development of the software; combined with a custom built machine learning-based prediction system. The software architecture was informed by the requirement for it to be an instantiation of the method, therefore that it should be capable of evaluation within the department's computer systems.

To the author's knowledge there has been no previous published research addressing the challenges of codifying neurosurgical-specific audit categories from free text. By combining highly specific rules-based information extraction with the weighted word counts of a machine learning component in a unique way, the method demonstrates a unique approach to creating applications that solve this codification problem.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Signature: …….

Print Name: ……..

Date: ……

Acknowledgements

This research and access to the datasets was approved by the Office of Research and Governance of Alfred Hospital and Monash University Human Research Ethics Committee (MUHREC).

First and foremost, I wish to express my sincerest gratitude to my supervisors Professor Frada Burnstein, Dr. Pari Delir Haghighi and Dr. Phil Lewis. I would like to specially acknowledge Frada, for not only being an incredible supervisor but also for being a remarkable and inspiring person. She has enabled me to begin my journey into research with her clear guidance and attention, but also with her continual support, encouragement and the opportunities she has given me. She inspires me to try to be like her - clear minded and insightful but also warm hearted, patient, and compassionate. I have been very blessed to have benefited from Frada's guidance.

Thank you Pari for taking so much time to guide me through this project, to read my work with a great deal of attention and to provide me with so many valuable insights in my research. I have been frequently amazed at how clearly you have seen things that need improving! I am also grateful to Dr. Lewis to find time for me in his very busy schedule and answer all of my questions with a lot of care.

My deepest appreciation goes to my best friend in life, my dear husband Christopher Palmer, who not only helped take care of family matters while I was studying but has provided unstinting assistance and insight into many aspects of my work – ranging from the information technologies utilised; to medical subjects; to fixing my English; and to having a genuine interest in what I have been doing. Thank you for your complete love and deep understanding.

I am also very grateful to my sister Mahsa, who very generously has taken care of our family while I was on this journey. I simply could not have done this without her help, and she has sacrificed a lot to have been such a support. Finally, I would like to also thank my dear parents, especially my mother Fatemeh Mohseni, who has always been encouraging and dedicated in helping us all in every step we take in our lives.

To the Most Generous

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The hypothesis that informs this thesis's research is that information extraction techniques, along with suitable machine learning (ML) algorithms represent reliable methods for generating audit codes from free text in the context of neurosurgery.

The UK's National Institute for Health and Clinical Excellence (NICE) defines clinical audit as: "... a quality improvement process that seeks to improve patient care and outcomes through systematic review of care against explicit criteria and the review of change. Aspects of the structure, process and outcome of care are selected and systematically evaluated against explicit criteria. Where indicated changes are implemented at an individual, team, or service level and further monitoring is used to confirm improvement in healthcare delivery" (*Principles for Best Practice in Clinical Audit*, 2002).

The audit process requires codified data for aggregation and analysis of patterns, this data typically comes from electronic health records (EHRs) or some other unique codified system required by a speciality. In clinical health records the codified system is typically structured around either a problem list or a diagnosis list.

Despite the obvious advantage of codified records they are often inaccurate, incomplete and not properly maintained (Kaplan, 2007); and free-form text remains a key component of electronic health records.

Researchers and developers of clinical information systems have used a range of technologies to try to achieve complete and accurate coded clinical data using post-hoc text processing. Amongst those technologies are Information Extraction methods and machine learning algorithms.

1.1 Problem Statement

Whilst a health organization's clinical information systems require structured data, unstructured free-form text is essential for clinicians to accurately describe the clinical encounter (Walsh, 2004). Therefore, a challenge exists to construct computing systems that can codify unstructured text.

The use of free-form text and a parallel lack of use of standard terms in text based electronic health records is extensive, and is a source of poor data quality and an inability to share data between systems, to construct decision support systems, and make secondary uses of data (Price et al., 2013). Free-form text is inevitably idiosyncratic and also very often incomplete, and lacking any coding structure does not allow for aggregation and analytical comparison.

However, free-form notes are almost always required to describe a patient's condition and treatment thoroughly, because abbreviated codes cannot adequately do this. The resulting "double entry" of then also having to code this information is often overlooked or seen as an unwelcome time-consuming extra requirement. Realistically codes cannot always be allocated at the time of entering notes – for instance a condition might remain un-diagnosed until later in the patient's treatment, but then not always retrospectively coded as further notes are entered.

Research indicates that despite the advantage of collecting structured data in the clinical software, clinicians value the narrative expressivity and workflow efficiency of entering free-form text (Rosenbloom et al., 2011). Systems designed to acquire structured data in real-time often have unnatural, inflexible, or inefficient user interfaces that place too much of a burden on busy clinicians, therefore it may be better to leverage computing technology to extract codified data from free-form clinical notes using post-hoc text processing (Ash et al., 2004).

A variety of technologies have been used to attain complete and precise coded clinical data using subsequent text processing techniques. Some have used natural language processing (Long, 2005; Meystre and Haug, 2006), others have used data mining and machine learning techniques

(Pakhomov et al., 2006; Wright et al., 2010). Rosenbloom et al. (2011) suggest that there is a need to develop hybrid systems that combine structured entry with later text-processing.

Whether codified data is obtained up front via the interface, or later via text processing, it remains a problem that data extracted from a single system may be incomplete. Wright et al. (2011) have combined data obtained from medications, laboratory results, billing codes, and vital signs and have designed rules for identifying a limited set of target conditions.

In summary, the adoption of organizational standards and computer assisted code entry systems have not resulted in complete adherence to a standard methodology and terminology, and even if such a system is properly used a strong requirement for the use of free-form text remains.

Auditing is an example of this situation: Auditing requires codified data, but the audit process takes place well after electronic health records have been entered – it is then too late to enforce the use of coding standards. While continued effort is required to ensure completeness as data is entered, there is an ongoing requirement to be able to assemble more complete codified data from whatever free-form text and other data sources that are found at audit time. The implementation of a customised text analysis and coding system is suggested especially for specialised in-house systems which use locally developed coding systems and free-form text, independently of the main electronic health system of the hospital.

This research aims to contribute to the urgent and growing need to understand the challenges of codifying free-form clinical text and to create effective systems that harness the best of information systems technology to solve these challenges.

1.2 Research Context

The Neurosurgical Department has an application that is used to describe the diagnoses and procedures performed for each patient passing through the department. This is independent of the main electronic records system of the hospital, and the data collected and terms used are highly

specific to neurological conditions and neurosurgical procedures. The medical terminology used is closely aligned to that published by the Royal College of Surgeons, in order to support regular reporting and surgical activity tracking.

Yearly auditing is performed using fine-grained data for selected common diagnoses, and coarse data for others. With this in mind, the data mapping algorithm is structured around the pattern of analysis derived from previous yearly audits undertaken by the neurosurgery department. Thus some individual diagnoses map to a single, high-level code, whilst others map to individual codes.

The structure of a record in the application is one of a code and an accompanying note, with as many records per admission as is required to properly code all of the diagnoses and procedures, though typically only one record exists per admission. A note is not required against a code, though it is expected that a note will appropriately qualify and amplify the code picked. Table 1-1 is an example of an admission record, admission code is de-identified.

Table 1-1 Example of an admission record

Admission Code	Date	Diagnosis	Notes
3301458954811	xx/xx/xxxx	Cranial>Trauma>Osseous Injury>Skull>Depressed>Open	Ped v car left frontal depressed fracture, GCS 3, ETOH

This study and access to the datasets was approved by the Office of Research and Governance of the hospital.

1.3 Research Aims and Objectives

This research aimed to develop a method for processing the free text entered in a neurosurgical department of a major trauma hospital; in order to improve the accuracy, coverage, and speed of extracting audit codes from the text.

Current techniques for codifying free text were evaluated to discover those most effective for the type of text used in the neurosurgical department; which led to a design that combined information

extraction techniques with machine learning in an innovative way. The design was realised as a computer-based method which was evaluated for accuracy and coverage using industry standard metrics, and which should be easy to incorporate as a working application into the department's workflow and computer systems.

The following research questions summarise the objectives:

1. What intelligent techniques can be used to develop a semi-automated audit code extraction method?
2. How can a code extraction solution be designed to be applicable to both the audit process and the initial data entry process?

These questions were addressed by following the principles of Design Science Research (Hevner et al., 2004) to construct a method that combines Natural Language Information Extraction and Machine Learning techniques.

1.4 Dissemination of the research

A part of the work described in this thesis has been published in a paper that was presented at the 26th Australasian Conference on Information Systems, entitled "Intelligent audit code generation from free text in the context of neurosurgery".

1.5 Structure of the Thesis

The thesis is structured as follows:

- **Chapter 1** describes the context of the research, including the problem the thesis addresses, the structure of data involved, and the research aims and objectives.
- **Chapter 2** introduces the topic of Information Extraction (IE), which is the basis of the techniques that the thesis explores. After a general explanation of the challenges of information extraction the chapter then highlights the particular issues faced in performing IE in the medical domain. Following this is a detailed description of the core techniques of IE – pattern matching, syntactical and semantic language analysis, and machine

learning methods. The chapter then describes the importance of pre-processing text, the usage of medical terminologies, and concludes with an analysis of how to apply IE techniques to the problems being explored by this thesis.

- **Chapter 3** describes the design techniques used in the research, beginning with an introduction to design science research methodology. The research output is a computer-based design, which is described as a *method* that combines the most effective techniques for solving the research problem, and which is instantiated as a software application and report. The architecture and components of the method are explained, and the evaluation process that was used to assess its accuracy.
- **Chapter 4** explores the research results with accompanying analyses.
- **Chapter 5** concludes the thesis, including a discussion of the unique contribution of the research. The limitations and proposals for future research directions are also explained.

Chapter 2

Background and Related Literature

This research is focused on using Information Extraction and Machine learning algorithms to improve extraction techniques for deriving audit codes from neurosurgical clinical textual data. Subsection 2.1 begins with an analysis of the unique characteristics of clinical text in relation to the extraction of codified information, including a presentation of related literature. Subsection 2.2 introduces Information Extraction, and Subsection 2.3 describes the various Information Extraction methods employed in the clinical domain, including domain-specific knowledge resources. Subsection 2.4 is a detailed discussion of Information Extraction methods, and Subsection 2.5 deals with text pre-processing in detail. The use of Medical Terminologies is covered in Subsection 2.6, and Subsection 2.7 wraps up with a discussion and analysis of the material covered. Subsection 2.8 concludes the chapter with a summary of the reviewed literature and the direction that this research will be taking.

2.1 Clinical Notes

Electronic health record (EHR) systems facilitate the reuse of clinical documents for purposes such as automated decision support, quality of care initiatives, and research (Demner-Fushman et al., 2009; Rosenbloom et al., 2006). A requirement for these processes is that clinical data should be structured, however a majority of useful clinical documentation is unstructured, abbreviated, and idiosyncratic - due to the fundamental requirement for clinicians to be able to enter information in a comprehensive and natural way (Rosenbloom et al., 2011). Clinicians most often take notes by hand or by dictation, and these notes are then transcribed as free-text data, but entering data up front in a structured way is seen as limiting.

Greenhalgh et al. (2009) describe the entry of free text paper records as being tolerant of ambiguity, supporting the complexity of clinical information, and by contrast entering data through structured mechanisms such as templates very often slows down and frustrates the recording of a clinical encounter.

Clinical staff must be able to communicate easily and thoroughly, this can only occur via natural informal communication methods: talking, writing, drawing on white boards, examining paper charts and printouts, reminder notes, etc. These are often the most effective and responsive ways of exchanging information, and studies have revealed that the most complex high-tech environments such as ICU rely heavily on these kinds of informal communication (Greenhalgh et al., 2009). Important data from these should be transcribed, but it would be impossible to interface adequately with the clinical environment via EHR oriented structured methods.

Rector (1999) describes a fundamental conflict between the needs of humans and computer programs: humans require to use flexible and expressive language while computer programs are generally designed to process formally structured data. The requirement for standardised systems requires pre-defined codes and controlled vocabularies, while an undistorted record of the data usually requires the expressive power of free text. In conclusion what is required is some kind of compromise between the two that does not compromise accuracy and completeness (Sager et al., 1994).

Even when note taking is included into an EHR system, entering these notes via a structured framework can be an impediment to the requirement for time-pressured clinicians to be able to be expressive or abbreviated in their text entries. Clinicians need to be free to use whatever language is natural for them (Rosenbloom et al., 2011). Rosenbloom has reviewed studies looking at the expressivity of free text notes compared to structured clinical documentation systems and has found that the natural language of free text notes is more complete and precise, and easier to understand (Rosenbloom et al., 2011). Rector (1999) also concludes that natural language will continue to be “richer in content and context” than any structured equivalent.

Improving the capture of structured information from clinical notes can be accomplished by either improving the ability for clinicians to interact with computer based systems - capturing structured information from their inputs at the time of taking notes; or by improvements in later interpretation of their clinical notes, or both.

Supporting structured clinical documentation entry will require many improvements in the usability of these systems - for instance by offering a greater variety of codified names so that a user can select a code that properly matches the concept entered in the clinical text. Numerous studies by informaticians over the last four decades have investigated what is required to improve structured data entry (Kohane and Uzuner, 2008), suggesting this is a better area to focus resources on rather than the extraction of codified data from narrative text, yet the use of narrative text continues to grow.

Other studies have found that clinicians will only accept processes that do not negatively impact their interactions with patients – accepting for instance the use of clinical encounter forms which enforce some structure but do not require data entry at the time of interviewing a patient - the form will be transcribed and converted to electronic form at a later time (van Ginneken, 2002).

Rosenbloom et al. (2011) reviewed and compared computer based documentation (CBD) systems with post-processing of free text records and found that post-processing is a valid option, and should be taken based on the need of the individual organization rather than any ideal about the best method. They additionally describe systems that perform post-processing over flexible CBD systems that allow for free text entry, to create post-hoc the required structured records - thus combining the strengths of both structured entry and post-processing.

In a hospital's medical record department there are professionally trained coders who are employed to code from notes, but this is generally for billing and financial purposes rather than clinical purposes, and a professional coder would not normally be employed to code the notes from a speciality such as neurosurgery. In any case research is dedicated to improving the consistency and reliability of coding through automation of the coding process using computer technology, which this research is also addressing.

Table 2-1 contains a few examples of the kind of free text physicians need to use in order to properly describe a patient's condition and the relevant circumstances that required his hospitalization. The information contained in the notes is abbreviated yet entirely adequate to

convey meaningful detail, but it must be coded for downstream statistical use by the hospital. The notes should all be mainly coded as CRANIAL:TRAUMA:TBI, meaning a traumatic brain injury, though a few other codes are also possible. If a physician was forced to enter just this coded information he could not convey any depth of information useful to understanding the patient's condition. Both types of information are required.

Table 2-1 Examples of unstructured and coded information

No.	Notes	Phrase	Meaning	Code
1	CHI, GCS 3, ped v train, ETOH	CHI	closed head injury	CRANIAL:TRAUMA:TBI
		GCS 3	Glasgow Coma Scale score 3	probable CRANIAL:TRAUMA:TBI
		ped v train	pedestrian struck by train	
		ETOH	alcohol	
2	ASDH R, burst frontal and temporal lobe	ASDH R	right sided acute subdural haematoma	CRANIAL:TRAUMA:SDH
		burst frontal and temporal lobe	frontal and temporal lobes of brain ruptured	CRANIAL:TRAUMA:TBI
3	gcs 3 pupils fixed dilated, brainstem reflexes negative, CT non viable brain injury	GCS 3	Glasgow Coma Scale score 3	probable CRANIAL:TRAUMA:TBI
		pupils fixed dilated	pupils fixed and dilated	CRANIAL:TRAUMA:TBI
		brainstem reflexes negative	no response to tests for brain activity	CRANIAL:TRAUMA:TBI
		CT non viable brain injury	Computerised tomography scan shows unsurvivable brain injury	CRANIAL:TRAUMA:TBI
4	R occipital contusion, R parietal SDH. Car vs tree, HS, LOC	R occipital contusion	Contusion of the right occipital lobe of brain	CRANIAL:TRAUMA:CONTUSIONS or CRANIAL:TRAUMA:TBI
		R parietal SDH	subdural haematoma in right parietal region	CRANIAL:TRAUMA:SDH
		Car vs tree	car hit a tree	
		HS	head strike	
		LOC	loss of consciousness	probable CRANIAL:TRAUMA:TBI

2.2 Information Extraction

Information Extraction (IE) is a process of retrieving specific targeted information from texts or speech and presenting them as fixed-format and unambiguous data. Like Information Retrieval (IR) it analyses the text for patterns using natural language techniques, but whereas IR will just return a series of documents matching a query, IE will return specific data from the documents (Cunningham, 2005). Unlike "full text understanding" which attempts to represent all of the

information in a text, Information Extraction is limited in its output: IE specifies in advance what is required – the semantic range of the output, the relations wanting to be represented, and the allowable data for each component of the output (Grishman, 1997).

Creating coded data from free-form text is comprised of Information Extraction and Mapping tasks. Information Extraction typically requires a pre-processing stage to clean the text and prepare it for processing, which then utilises various techniques to categorise the text into entities of interest: a task called Named Entity Recognition (NER) (Meystre et al., 2008). Those named entities (NE) can then be mapped to their corresponding concepts in standard terminologies and used for creating codes.

Pre-processing includes document structure analysis, spell checking, sentence splitting and word tokenization, part-of-speech tagging, word sense disambiguation, and parsing to identify words of interest (Demner-Fushman et al., 2009).

NER takes the words parsed from the free form text and matches them to their corresponding named entity, taking into account contextual features like negation, temporality, and event subject identification in order to accurately classify and interpret the words. Examples of contextual features are negation (e.g. *without* elevated temperature); temporal information (e.g. *previously* admitted for pneumonia); family history (e.g. *family history* of breast cancer) and event subject identification (e.g. *his mother had* diabetes), and modifiers denoting that the event may not have actually occurred (e.g. *possibly exposed to* Ebola virus, admitted for observation).

In the clinical environment named entities are typically categorised as symptoms, investigations, test results, diagnoses, prognoses, drugs, treatments and procedures, and outcomes of treatments and procedures (Wang and Patrick, 2009).

Various paths to NER can be found in the clinical literature - they generally use one or a combination (hybrid) of three approaches: rule-based, dictionary-based, and machine learning-based approaches (Krauthammer and Nenadic, 2004). These can all be incorporated into Natural

Language Processing (NLP) systems - which is a popular approach used by software tools that have been developed to deal with the codification of free-form text (Agah, 2013).

Research shows that these various systems perform relative to the complexity of the task and desired outcome - for instance while an NLP based system may deal well with descriptive language it needs additional components to deal satisfactorily with structured items such as laboratory test results. Therefore it remains difficult to reach conclusions about the most effective tool, and currently there is no general uptake of these tools in clinical practice (Stanfill et al., 2010).

Unambiguous data means that IE requires that ambiguities in the text are resolved. These include tasks such as converting temporal expressions such as "tomorrow" into an actual date; and disambiguated words which can have several different meanings (polysemous words) based on context – a commonly used example is *bank:* it can refer to the margins of a river, a place to put money, or a way of describing something you can rely on.

The definitions of the tasks involved in Information Extraction were described in the late 1980s and 1990s by the Message Understanding Conferences (Grishman and Sundheim, 1996). The MUCs crystallised five general extraction tasks (Cunningham, 2005):

Table 2-2 MUC Extraction Tasks

Extraction task	**Explanation**
Named Entity recognition (NE)	Finds and classifies ***entities*** - names, places, events, etc.
Coreference resolution (CO)	Finds ***identity relations*** between entities - which entities and references (e.g. pronouns, nouns) refer to the same thing
Template Element construction (TE)	Adds ***attributes*** to named entities - descriptive information, some of which are obtained via NE and CO, others from domain rules
Template Relation construction (TR)	Finds ***relationships*** between Template Element (TE) entities
Scenario Template production (ST)	Describes what ***event scenarios*** Template Elements (TE) and their Relationships (TR) participate in

Coreference resolution refers to different pronouns and nouns describing the same entity into the one named entity – e.g. *John* said… *he* said – both being *John* (anaphora resolution); and resolving variations of a company name such as "McDonald's Corporation", "McDonald's", "Maccas", and "McD's" (proper noun coreference identification).

A template is essentially a database record, which has specific fields of interest in addition to the entity name – such as the type of entity, the location of the entity, other aliases, titles, etc. which are useful to identify and make use of the entity in relating the data being processed. A scenario event is the ultimate output of IE, it is information about the sequence of events that an entity partakes in – such as a person resigning from one company to start work in another one, or a stock market trend, or the outcome of a surgical operation.

Named Entity extraction and coreference resolution are the primary data extraction tasks, the construction of templates and scenarios are to do with deriving information from the data, and the more complex the process is the more imprecise the results tend to be; or if targeting precise results then the more limited will be the coverage of the results. These factors are referred to precision and recall – a very precise method will recall less, whereas when recall is maximised there is less surety about the precision of the data (Cunningham, 2005), and they are explained in detail in section 4.3 in this thesis. Depending on the complexity of the domain and the inputs required by further processing steps it may be sufficient to obtain just the resolved named entities from text.

2.3 Information Extraction in Clinical domain

Clinical information extraction is defined as information extraction performed on clinical text (Roberts, 2012). Clinical natural language processing and information extraction are often treated as a distinct subset of the tasks, as clinical texts have specific characteristics.

Sentence boundary detection is more difficult in sentences containing abbreviations, and terse sentence structures with a lack of punctuation or narrative flow, and the use of medical titles, lists, etc. (Nadkarni et al., 2011).

Even within a domain there will be a **wide range of text entry styles** depending on the practitioner – with no uniformity in the text – some being prose-like, others being highly telegraphic and ungrammatical – it becomes **difficult to establish rules** for processing the text (Spyns, 1996).

Tokenization depends on finding non-alphanumeric characters such as full stops, but this has to take into account the use of these characters in drug doses and other measurements, and in chemical and drug names. Single characters or combinations of very few letters may in fact be significant abbreviations, whereas others may not be – in any case in an ordinary NLP task they would be discarded. Some of these may be placed in unusual places compared to standard texts – such as "?" before a word to indicate uncertainty of the following diagnosis – in an ordinary NLP task "?" would indicate the end of a sentence.

Tables of data from laboratory results and the like may be pasted into text, resulting in many non-prose and non-character tokens and strange formatting, which means that tokenization often has to be purpose built to cope with these when determining sentence boundaries etc. Plain text may be formatted as a table but not within a standard table structure (a pseudo-table) – this can be difficult to process: a standard table structure can be used by a parser to understand the table elements whereas a pseudo-table has no clear information that allows the table to be properly understood.

Medical texts are **frequently qualified with modifiers** suggesting uncertainty or possibility, negations - using phrases like "indicative of", "possible", "less likely" (Nadkarni et al., 2011). Some of these may be abbreviated – e.g. the use of "?" described above, and "nad" meaning "no abnormality detected", which is often used in relation to a preceding text.

Entity recognition is made more difficult by variations in word order and spelling (e.g. "C7 right superior articular facet #" and "#R C7 sup art facet"), the use of synonymous and polysemous words, and abbreviations and acronyms - which can be specific to a particular institution (Xu et al., 2009), or area of practice, or even to a practitioner.

Abbreviations themselves are often polysemous and at least a third of them have more than one sense (Liu et al., 2001). For example the use of "art" as an abbreviation – it can mean "articular" or "artery" – so context has to be assessed – "sup art" would translate to "superior articular" – a part of the spine, whereas "fem art" would translate to "femoral artery" – a major blood vessel in the leg. Another example of the use of very abbreviated terms combined with other abbreviations are the terms "ped" and "v": "ped" can mean "pedestrian", "paediatrics" or "patient examined by doctor" (among other official abbreviations), and "v" could mean a measurement (i.e. 5) or "versus" – so "ped v car" would be disambiguated to "pedestrian" and "versus", whereas "ped haem GCS V" would indicate a "peduncle haemorrhage" (in this case likely a specific practitioner's abbreviation) and a measurement on the Glasgow Coma Scale.

The very **dense use of terminology** and the rapidly changing corpus of terms used - with new terms being constantly introduced - causes problems of identification. In addition to the seemingly unlimited number of technical terms, general language terms are often given different meanings specific to the medical domain (Fisk et al., 2003).

Compounding these problems, **misspellings are common** – these can be due to the large number of second-language learners of English working in the health professions, and also to there being few built-in spell checkers in the medical data entry systems. So in addition to the many technical terms on top of general English terms an NLP system has to deal with misspellings.

Data coming from these systems is often anonymised to overcome ethical constraints, and in doing so the **relationships between records are obscured**, making it difficult to spot coreferences and to build up a complete picture of the information that should be extracted for a specific entity.

Additionally, **temporal information is presented in a domain-specific way** – there may be little to indicate when or if an event has occurred – for example "? Glioma on CTB" may indicate that computerised tomography of the brain has been conducted and revealed a possible Glioma, or it could mean that a CTB needs to be performed looking for a possible Glioma - in any case the temporal information is missing (Gaizauskas et al., 2006).

Meystre et al. (2008) conclude that these kinds of problems create new challenges for NLP techniques whereas other researchers conclude that traditional techniques including dictionaries, spelling correction algorithms and morphological routines are unlikely to succeed with these sorts of texts, and that a corpus-driven approach is required to construct application-specific dictionaries and routines (Lehnert et al., 1995).

If it is possible to define domain and problem specific terms of interest, such as identifying diagnoses pertinent to neurosurgery, then algorithms and dictionaries can be purpose built for the task of finding terms of interest, and other features of the text can be given lesser priority - though still assessed for patterns that can contribute to the extracted information.

2.4 Information Extraction methods

Meystre et al. (2008) conducted extensive reviews of the most recent developments (post 1995) in clinical IE, where they reviewed all of the current techniques ranging from pattern matching, contextual feature detection, to systems that combined semantic and syntactic analysis, and how these were implemented by various applications and used for contexts ranging from decision support, surveillance, and for research. Demner-Fushman et al. (2009) reviewed current clinical NLP systems and how they can be applied to decision support, and Stanfill et al. (2010) reviewed the use of NLP in the field of automated clinical coding and classification.

A summary of the findings of these reviews are that:

- A large volume of accurately annotated clinical text corpora are a main requirement for research (for training and testing of IE systems a lot of text with correct associated codes is required)
- To ensure accuracy and remove biases annotation should be carried out by more than one reviewer, and be cross checked
- More competitive challenges to perform information extraction from clinical texts will stimulate advances in the field
- Performance of automated clinical coding and classification systems is relative to their design specificity for solving the complexities of particular situations, therefore these systems cannot be made generalizable without sacrificing performance
- Applications are rarely applied outside of the research situation they have been developed in, mostly because of generalizability and scalability issues
- Because of the research focus on solutions for the various challenges, the work done has focused on a limited number of subjects, namely the processing of discharge summaries and radiology and pathology reports
- Future research will be required to establish whether systems can be retargeted to solve problems in other domains and still compete with specifically designed systems
- There is little research into whether the systems reviewed are actually implemented and what success they have after implementation
- Additional study is needed about how to measure the level of performance required for a system to be useful in real-world clinical tasks

The following section describes the various methods used for clinical Information Extraction.

2.4.1 Pattern Matching

Pattern matching techniques (McNaught and Black, 2006) such as regular expressions and the use of dictionaries of synonyms can perform well, especially if the text either conforms to a limited set of expressions such as in a report of test results, or if the text is very telegraphic and uses abbreviations repeatedly – this kind of text is not very amenable to more sophisticated sentence structure analysis (syntactic modelling). At its most basic, pattern matching uses regular

expressions to look for patterns in the text that identify an entity, a further development is to identify consistent elements in text that can be assembled according to defined rules to identify an entity – this is the basis of semantic modelling. More developed techniques take into account semantic and syntactic relations in order to understand the entity in relation to negations and other important meaning modifiers.

Turchin et al.(2006) use regular expressions to extract blood pressure and antihypertensive treatment information. Friedlin and McDonald (2006) developed the REgenstrief eXtraction (REX) tool – which uses targeted NLP and pattern matching to extract diagnoses such as congestive heart failure from chest x-rays. REX has a structural analyser that recognises the main sections of a report, and then uses multiple regular expressions combined with rules to discover word associations, synonyms, and qualifiers in a sentence. Friedlin and McDonald (2006) note that general purpose NLP tools linked to SNOMED or UMLS, though suitable for extracting detailed information from large bodies of text, are not suited for getting useful information from a targeted concept space.

Bashyam et al. (2007) describe a module they developed for MMTx - the java version of MetaMap - that extracts UMLS concepts using pattern matching. It is considerably faster than MMTx and they suggest it can be used to return single UMLS concepts in real-time text-mining applications.

Napolitano et al. (2010) describe a good example of the suitability of pattern matching using regular expressions for extracting information from structured text. Looking for variants of descriptions of a Gleason score, they were able to correctly classify 98% of the texts. The system they developed was capable of extracting the Gleason score from a test set of 915 documents after only 4 hours of coding to fine tune the extraction, whereas manual entry of the data took 30 hours.

Gold et al. (2008) describe a method for extracting medication information from discharge summaries, by using parsing rules expressed as a set of regular expressions and implementing an editable drug lexicon. They observe that medications and medication event information have a

limited number of variations and language structures – as such they constitute a "regular language" rather than the more variable "context free language" typically used in NLP applications. They were able to model the language through manual observation and constructing parsing rules – using regular expressions to identify drugs, possible drugs, and context rules, and the dictionary to check for synonyms and misspellings. Their method used a relaxed approach to identifying drugs and drug events – not insisting on drug names and dosage for example to include a drug, but looking at surrounding text to indicate that the mention of the drug constitutes an entry: they give the example of "Pt will remain on antiseizure medications" as a valid reference for inclusion (Gold et al., 2008).

In summary, pattern matching is suited to text that has a consistent structure, where terms are used repetitively, and also where the text is minimal or the language used is terse and lacking the extra clues that would make it suitable for a deeper language analysis. However, if the text is more natural or has a greater frequency of contextual clues it is amenable to standard natural language analysis – and it then becomes necessary to process the text to get a complete knowledge of how phrases are constructed and combined – syntactic data, and how they should be interpreted – semantic understanding, so as to not miss vital information about the meanings of the text.

2.4.2 Syntactic/Semantic

The majority of the systems used in clinical text processing combine syntactic and semantic analysis, and one of the most widely used and adapted systems is MedLEE (Meystre et al., 2008).

MedLEE (Medical Language Extraction and Encoding system) is a mainly semantically driven NLP system created by Friedman at al. (Friedman et al., 1994) - it was developed to extract information from clinical narrative reports, as a component of an automated decision-support system, and to allow natural language queries. MedLEE was generalizable enough to be then utilised by another institution, wherein only a small performance decrease was observed. After some adjustments were made it performed as well as in the original institution (Hripcsak et al., 1998). MedLEE has been adapted to extract UMLS concepts and codes, with corresponding

modifiers, from medical text documents – correctly identifying 83% of the classifiable concepts (Friedman et al., 2004). Freidman et al. give an example of a sentence "Status post myocardial infarction in 1995" - where MedLEE delivers the UMLS concept "post msi" as being more accurate than the concept "myocardial infarction", along with a date modifier with a value of 1995. The "post msi" concept translates to the corresponding UMLS code C0856742.

MedLEE is comprised of the following modules: pre-processor, parser, error recovery, phrase regularisation and encoding.

- The **pre-processor** divides the text into sections, paragraphs, sentences, and words and uses a lexicon to determine their canonical forms. An abbreviation table is used to define abbreviations, and some word sense disambiguation is performed based on contextual rules.
- The **parser** determines the structure of the language and the correct interpretation of the language for each sentence. It makes use of a grammar that includes syntactic and semantic rules, and its output is a structured list where the first element is an information type, the second element a value, and the remaining elements are modifiers. For example "Status postmyorcaridal infarction in 1995" would be structured [problem,'myocardial infarction', [date,'19950000'], [status,post]].
- **Error recovery** attempts to obtain a result if the initial effort failed, by various techniques like skipping words and segmenting the text.
- **Phrase regularization** finds instances of non-contiguous phrases that if reorganised would equate to a lexicon defined multi-word phrase, reorganising them into the appropriate multi-word phrase assigning them a degree of certainty to the match. For example the parser dealing with the phrase "spleen was enlarged" would yield an initial value of [problem,enlarged, [bodyloc,spleen], [certainty,'high certainty']], but after phrase regularization it would be rearranged to match the multi-word phrase "enlarged spleen" as [problem,enlarged spleen, [certainty,'high certainty']]. Additionally, phrase regularization might add domain-specific information to the output - for example if the

domain is cardiology and the sentence contains “infarct” with no specific site, it would be rendered as “myocardial infarction”.

- **Encoding** makes use of an encoding table to add UMLS (or other codes as specified), to the output form.

SPRUS (Special Purpose Radiology Understanding System) (Ranum, 1989) was an NLP application developed by the Medical Informatics group at the University of Utah. SPRUS was the first of a series of applications developed, and was used to derive semantic information from radiology reports by referencing a diagnostic knowledge base.

SymText (Symbolic Text processor) was a later development by the Medical Informatics group – it combined a syntactic structural analyser, transformation into concepts according to targeted semantic structures, and resolution of relationships between the extracted concepts and their meanings using Bayesian (belief) Networks (BN) for semantic analysis (Haug et al., 1995). The output of SymText were codes to describe findings, diseases and devices, and it was used at the LDS Hospital in Salt Lake City Utah to semi-automatically code admission diagnoses into ICD-9 codes (Haug et al., 1997).

MPLUS (Medical Probabilistic Language Understanding System / M+) is a further evolution from SymText – its difference is in the size and modularity of the BN used, and the use of a bottom-up chart parser with a context free grammar. With MPLUS syntactic and semantic analyses are interleaved, which contrasts with most other NLP systems that perform the semantic analysis after the syntactic parse (Christensen et al., 2002). The semantic model can trained to work with new domains and so adapted to new applications - MPLUS has been used variously to codify chest radiological reports, CT scans of the brain, admissions in a Level 1 trauma centre (Day et al., 2007), and complaints in syndromic (outbreak) surveillance (Chapman et al., 2005).

UMLS MetaMap was described in 2001 by Aronson (2001) developed at the National Library of Medicine (NLM) - initially to enhance biomedical text retrieval (Aronson and Rindflesch,

1997). MetaMap is a highly configurable NLP program that maps biomedical text to concepts in the UMLS.

MetaMap uses a knowledge-intensive approach based on symbolic, natural language processing (NLP) and computational-linguistic techniques, to map biomedical text to concepts in the UMLS. It is highly configurable, including in its data options - allowing a choice for the vocabularies and data model to use; its output options - in determining the nature and format of the output generated; and its processing options - choosing which algorithmic computations should be performed by MetaMap (Aronson and Lang, 2010).

MetaMap, originally developed in Quintus Prolog, and a Java version of it called MMTx (MetaMap Transfer), have been used to extract information from many clinical documents. For example Meystre and Haug (2006) used MMTx to automatically generate problem lists - where it showed reasonable performance. MetaMap has been used in many approaches to support recognition of named entities related to cancer (Spasić et al., 2014). MMTx has since been discontinued as its rationale for development - which was to allow for Prolog license-free development by MMTx users, has largely not been adopted.

cTAKES (Clinical Text Analysis and Knowledge Extraction System), (Savova et al., 2010) is an open-source NLP system developed at the Mayo Clinic using Apache UIMA and Apache OpenNLP natural language processing toolkits. It processes clinical notes using a variety of rule-based and machine learning methods to identify clinical named entities by using UMLS or other dictionaries - classifying the entities as medications, diseases/disorders, signs/symptoms, anatomical sites and procedures. Named entities have properties to locate them in the text and identify them by ontology mapping code, subject (patient, family member etc.), and context (conditional, degree of certainty, negated or not negated etc.).

cTAKES is being actively developed by its community of users, with a goal to be a world-class scalable, comprehensive, modular, extensible and robust NLP system in the healthcare domain (Savova et al., 2010). Imler et al. (2013) assessed the accuracy of cTAKES to process and extract

clinical concepts from colonoscopy and their linked pathology reports. Their result showed that cTAKES was able to accurately (84-97%) capture key clinical concepts.

HITEx – the Health Information Text Extraction tool, was developed by Harvard Medical School and Brigham and Women's Hospital. It is incorporated as the NLP module in a collection of software published by i2b2 (Informatics for Integrating Biology and the Bedside), called the i2b2 Hive – which is an open source modular system created by consisting of a core set of modules and various optional modules, plugins, and a web client. Zeng et al. (2006) describe how HITEx was built using many components from an open source text processing system called GATE – General Architecture for Text Engineering, developed by the University of Sheffield. GATE includes a set of NLP modules known as CREOLE – a Collection of REusable Objects for Language Engineering, which form the basis of much of HITEx. For example, to extract smoking-related diagnoses a GATE Section Splitter, Section Filter, Sentence Splitter, Sentence Tokenizer, Part-of-Speech, Noun Phrase Finder, UMLS Concept Mapper and Negation Finder modules are applied sequentially. There are many technologies at work here, for instance the Classifier is a Support Vector Machine that uses single words as features.

GATE is still very much actively developed and used, recent commercial customers include MedCPU, Ontotext, and the UK Press Association. GATE has been used as the basis of very many research designs ("GATE," n.d.).

Table 2-3**Error! Reference source not found.** summarises these applications. Most of them use UMLS for their ontology, and their output also largely is standardized to the UMLS common user interface (CUI). cTAKES and HITEx are readily adaptable to new domains, but in order to make use of any of these applications the target systems must be configured to use both the incoming ontologies and outgoing encoding systems, and the language structure of the clinical data must be amenable to the NLP algorithms used by these applications. Furthermore, incorporating the programming languages, technical proficiencies, and workflows required to use these applications is a major commitment.

The non-standard ontology and abbreviated language employed in the neurosurgical department is not amenable to working with these applications, and the requirement for targeting specific components of the text would require a lot of fitting. For example, a trial of cTAKES over the neurosurgical notes produced very many spurious results, such as CHI (closed head injury) being interpreted as a reference to a Chinese medicine energy concept. Where words were correctly identified they were often words that were not requiring identification (such as the cause of an accident), or too many possibly competing terms were offered. A conclusion was reached that a specific, targeted approach was required to efficiently process the neurosurgical department's notes.

Table 2-3 NLP applications summary table, adapted from Doan et al. (2014)

Application	Clinical Domain	Programming Language	Framework	Ontology	Encoding of Output	Creators	Licensing
MedLEE	Radiology, Mammography, Discharge summary	Prolog		Internal Medical lexicons (MED)	UMLS's common user interface (CUI)	Columbia University	Closed source, but commercialized
SPRUS, SymText, MPLUS	Radiology, Admission diagnoses	LISP, C++		UMLS	ICD-9	University of Utah	Closed source
MetaMap	Biomedical texts, Mapping to UMLS concepts	Perl, C, Java, Prolog		UMLS	UMLS's CUI	National Library of Medicine	Not open source, but available UMLS Metathesaurus License
cTAKES	Clinical notes, Discharge summaries, Named Entity classification	Java	UIMA	UMLS and trained models	UMLS's CUI, RxNorm	Mayo Clinic, IBM	Open source Apache
HITEx	Family history, Smoking studies, Principal diagnoses	Java	GATE	UMLS	UMLS's CUI	Harvard Medical School	Open source i2b2

2.4.3 Machine Learning

Recent medical NLP systems almost always incorporate some machine learning methods, as a processing step or as the basis for the system. Many of the best systems using machine learning methods were developed in response to so-called "shared challenges", which are competitions set up by medical research and training institutions (Pestian et al., 2007). These organizations have access to large volumes of deidentified, cleansed, and approved data that represents problems faced in medical records coding, and the organizations run the challenges in order to advance research into computational medicine (Uzuner et al., 2011). Large volumes of this sort of "gold

standard" data are a requirement to properly train machine learning systems, which are at the forefront of the research. Following are described two of these challenges and conclusions drawn about how machine learning techniques might be incorporated into the method being developed here.

The International Challenge on Classifying Clinical Free Text Using NLP

In 2007 the Computational Medicine Centre (CMC) of the University of Cincinnati created the International Challenge on Classifying Clinical Free Text Using Natural Language Processing - a shared task challenge to develop systems that could process free-text radiological reports and assign one or two labels from a set of 45 ICD-9-CM codes (ICD-9-CM: The International Classification of Diseases, 9th Revision, Clinical Modification). For this the CMC supplied 1,954 de-identified radiology reports, together with corresponding manually-coded ICD-9-CM codes. Pestian et al. (2007) describe the project as very well received, especially with regard to the evaluation metrics, and 44 research teams took part – a significant response to the challenge.

As a result of the challenge there is research literature describing many of the systems created, and while the majority of the systems constructed were rule-based using hand-crafted expert rules, three of the top four highest scoring systems incorporated machine learning components to replace some aspects of the hand-crafted approach.

Farkas and Szarvas (2008) placed first in the challenge with a hybrid hand-crafted and machine learning system, which they demonstrated would score almost as well as a purely hand-crafted approach. They suggested that incorporating machine learning components would be optimal in a system that needed to deal with thousands of codes (rather than the 45 required by the challenge), as such a system can be developed quickly with less human effort. There were several parts to their system: They automated the construction of expert rules for labelling from the published ICD-9-CM labelling specification, and they determined rules for not over-coding or under-coding by utilizing machine learning.

Over-coding occurs when the system labels a diagnosis *and* also its constituent symptoms, and must be prevented - as the purpose of the ICD-9-CM labelling challenge was for billing, and it is

fraudulent to bill twice. The rule regarding over-coding is that if a diagnosis can be found in the text then it should be labelled, and any further text describing symptoms relating to the diagnosis should be discarded. To automate this Farkas and Szarvas (2008) trained a C4.5 decision tree classifier for each symptom label so that the classifier learned to distinguish false positive symptom labels (where an appropriate diagnosis can be derived or already exists) from true positive ones (where no equivalent diagnosis exists and therefore the symptoms can be coded). While this produced only a 1.5% improvement over a hand-crafted system it effectively demonstrated that a machine learning approach can replace a hand-crafted one.

Under-coding occurs when an infrequently used, or idiosyncratic, or synonymic, or abbreviated term cannot be identified by the hand-crafted rules, and so a label is not generated, representing lost revenue. Farkas and Szarvas (2008) again trained a C4.5 decision tree – teaching it to recognise classifiable phrases by learning from repeated occurrences of words, or pairs or triplets of words, that were equivalents of the known terms. They were able to extend the accuracy of the system by 5%, thereby again demonstrating the effectiveness of a machine learning approach.

Finally, Farkas and Szarvas (2008) used the data obtained by their classification training to enhance the dictionary that can be used by a hand-crafted system, and were able to demonstrate that a hybrid hand-crafted and machine learning system could perform almost as well as a purely hand-crafted system, but having the advantage of automation and accelerated development time.

Goldstein et al. (2007) evaluated three systems: one applying the open source search engine Lucene; the second using BoosTexter - a boosting algorithm based on n-grams; and the third a set of hand-crafted rules that captured lexical elements. They took these approaches on the basis that the minimal text being used in the radiology reports could be processed by systems that look for individual words or small strings of words. Lucene compares documents to one another in order to classify them based on the frequency of the use of terms using “term frequency-inverse document frequency” (td-idf) classification. BoosTexter uses a machine learning algorithm called boosting which cycles through text improving the classification with each cycle, but does not take account of semantic information. Their third system was a hand-crafted system that could take

full account of semantic information: finding negations and uncertainty to eliminate text (preventing over-coding), using synonyms and disease descriptions to include text (preventing under-coding). They concluded that a simple hand-crafted rule-based systems utilising lexical elements and semantic information outperformed more complex systems, and their submission placed second in the challenge (Goldstein et al., 2007).

Suominen et al. (2008) placed third. They performed a multi-label classification task against the text – testing each of the 45 possible labels via two classification techniques in order to eliminate labels from contention until only the most likely correct label remained. They would first pass NLP pre-processed text into a Regularized Least Squares (RLS) classifier, but if that suggested an empty or impossible combination of codes then the output of that would be passed to a RIPPER rule induction-based classifier, which they found would satisfactorily complete processing when RLS could not.

As their response to the challenge, fourth placed Crammer et al. (2007) described a multi-component coding system. Firstly, an NLP based learning system which was trained on the text and the descriptions of the official codes assigned to it – finding all the various configurations of text in reports in relation to code descriptions of the 45 codes. Secondly, a rule based system that assigned codes based on the relationships discovered by the learning system. Finally, a system that mimicked the guidelines that a medical coder follows when assigning codes, to optimise the correct code choice based on coding policy. They implemented the components in a cascading manner based on the accuracy of each component – using the coding policy first as it was the most accurate when it could match a code, then the rule based system, and finally feeding remaining inconclusive results into the NLP based learning system.

Aronson et al. (2007) responded to the challenge by combining several technologies including the National Library of Medicine (NLM) Medical Text Indexer, Support Vector Machines (SVM), a k-Nearest Neighbour (k-NN) classifier and a pattern-matching method - using a modified ensemble method based on stacking. They reported scores considerably higher than averages they compared against.

The CMC challenge organisers review of the submissions suggested that the use of a machine learning classifier (the statistical approach) was not as important to success as processing the text for negations, synonyms, and hypernyms; mapping linguistic rules linking text to codes (the symbolic approach); with the structure of the UMLS a contributing factor (Suominen et al., 2008). The studies described here likewise all take into account the requirement for very specific analysis and fitting of the systems – hand-crafted methods, and the best of them use a hybrid approach.

The i2b2 NLP challenges

i2b2 (Informatics for Integrating Biology and the Bedside) is a National Centre for Biomedical Computing based at Partners HealthCare System in the United States. The i2b2 are developing informatics frameworks to enable clinical researchers to make use of clinical data for discovery research in the design of targeted therapies for patients who have diseases of genetic origin. Core to their frameworks are a collection of software modules named the i2b2 Hive, which can be used to develop systems that utilise patient data for research – the NLP component of the Hive, called HITEx, is discussed in more detail in the preceding section of the thesis. i2b2 also develop disease-based Driving Biology Projects (DBP's) to serve as test systems for core disease research. Current DBP's are Autoimmune/CV Diseases and Diabetes/CV Diseases, past DBP's have been Airways Diseases, Hypertension, Type 2 Diabetes Mellitus, Huntington's Disease, Major Depressive Disorder, Rheumatoid Arthritis and Obesity.

As much of the data used in the DPB's is in the form of unstructured text, the i2b2 have sought to unlock that data by providing sets of deidentified data in a series of NLP challenges: The 2006 Deidentification and Smoking Challenge, the 2008 Obesity Challenge, the 2009 Medication Challenge, the 2010 Relations Challenge, and the 2011 Coreference Challenge. The current challenge is the 2014 Deidentification and Heart Disease challenge.

As a result of these challenges hundreds of journal and conference articles have been published by the research community, and the top performers in some of the challenges are summarised here in relation to the use of Machine Learning in NLP.

The 2010 Relations challenge used deidentified annotated reference standard data supplied by Veterans Affairs (VA) Salt Lake City Health Care System in partnership with i2b2. It was a three-tiered challenge that studied:

- extraction of medical problems, tests, and treatments (concept extraction)
- classification of assertions made on medical problems (assertion classification)
- relations of medical problems, tests, and treatments (relation classification)

The challenge culminated in a workshop, where the top 10 performing medical problem concept extraction systems all used some form of machine learning: Five used supervised machine learning, two used semi-supervised systems, and three used a hybrid of machine learning and shallow rule-based systems. The top 3 concept extraction systems were a semi-supervised and then two hybrid systems. The relation classification top 10 systems consisted of eight using supervised machine learning, one semi-supervised, and one hybrid system (Uzuner et al., 2011), with the top 3 being a semi-supervised, then hybrid, and then a supervised system.

The concept extraction systems performed best on inexact evaluation using textual features, and there was no significant advantage obtained by adding resources such as UMLS to try and deliver more exact results. The most effective concept extraction systems used conditional random fields (CRFs), which is a technique for classifying words according to their parts of speech but also to take into account nearby words that may modify the understanding of the word. CRFs are a type of logistic regression technique that models sequential labels.

The most effective assertion and relation classification systems utilised support vector machines (SVMs), often in conjunction with contextual information and dictionaries to define uncertainty, negation, and family history, or as a final step in a system that consumed the output of rule-based systems. A common approach was to identify the large number of concept pairs that could be easily separated out, leaving only those concepts that needed further processing.

The general trend in the 2010 challenge was to design systems that were ensembles of complementary approaches, with rule-based systems providing input to a machine learning

component, or rule-based post processing of machine learning output. Although the systems required reference to highly specific domain knowledge, the general approach taken in solving them was disease-agnostic and linguistic, which means that the techniques can be adapted and applied to other problems, not necessarily only in the clinical domain.

In conclusion, analysis of these challenges shows that machine learning systems contribute to the rapid development of systems, capable of replacing in some part a rule-based system (such as named entity recognition), but at the expense of a requirement for large and properly structured training sets. Although suitable training sets are sometimes made available in research challenges like the one described here, they are difficult to obtain in the real world of clinical text-based record systems, and obtaining the required training data remains a major limitation of machine learning.

For the purposes of this research it was important to test the ability of machine learning systems to identify any remaining significant language features that had not been previously been described as rules by the system expert. In the absence of properly annotated data, it was determined that the ML output could be tested against the incoming audit code of a note, with the expectation that where sufficient examples existed then statistically significant patterns would be identified.

2.5 Preprocessing text

In order to produce the features for subsequent tasks of Information Extraction, some preprocessing steps need to be taken to transform the text into a consistent form and remove noise, redundancies, incoherence and bias (Iavindrasana et al., 2009).

These tasks include:

- **Sentence boundary detection:** This task is to identify the beginning and end of a sentence. In normal text punctuations such as a full stop signal a sentence boundary but in the clinical domain there can be no boundaries in the case of unpunctuated text; or

unusual characters used for boundaries; or abbreviations, values and titles that contain punctuation characters (for instance 'c.s.f.,' 'Dr.').

- **Tokenization:** This step consists of breaking a sentence to individual units including words, punctuations, symbols and numbers. The complexity of this task in clinical domain is due to tokens containing characters that typically define token boundaries, such as slashes and hyphens ('C4/5', 'GCS-V').
- **Part-of-speech (POS) tagging:** Once text is tokenized, words or multi-word terms should be identified and parts of speech, such as nouns, verbs, and adjectives should be assigned to them. A word can be associated with more than one part of speech (e.g. can be a noun or an adjective) so context can be used to determine the correct assignment.
- **Shallow parsing (chunking):** The identification of phrases ("chunks"), where it is important to keep words together to maintain their correct meaning – for example a noun phrase (NP) such as 'subdural haematoma', verb phrases (VP) consisting of a verb and its auxiliaries, and preposition phrases (PP). These are most likely more useful to understanding the text than classifying it only as a collection of individual words. A top-down approach known as **chinking** can be performed, where instead of adding words together, superfluous words ("chinks") are removed to get meaningful phrases.
- **Case:** Often pre-processing will force all terms to lower case so the task of matching is simplified, but the use of upper case and context is often the only way to determine that a word is an acronym – for instance CAT = 'computerised axial tomography'. To cope with this pre-processing may need to accept case, or terms in upper case may need to be translated before reducing to lower case, or the pre-processing may need to take account of all upper case words and reduce the rest of them into lower case.
- **Morphological decomposition:** In order to identify compound words as much as possible by one pattern, suffixes can be removed (*stemming* – for example "having" reduces to "hav"); or preferably words can be reduced to their root meaning (*lemmatization* - for example "having" reduces to "have"). Variants and synonyms of words can be reduced to one version of them via a dictionary. In the clinical area care must be taken not to reduce a word into an ambiguous root – for example "spinous" and "spinal" would lose their meanings if reduced to "spine". Further, the widespread use of abbreviations in medical

text means that these may need to be identified and expanded before decomposition, so that their meaning is not lost by other words being reduced to the same pattern when decomposed. If a medical domain is specialised and therefore has a relatively limited vocabulary, and the objective is to identify certain key words, then decomposition may not be necessary.

- **Problem-specific segmentation:** segmenting text into meaningful groups, such as Chief Complaint, Diagnoses, Past Medical History, Procedures, and Tests. In typical abbreviated medical entries entire sentences may be able to be classified as pertaining to a problem.
- **Spelling/grammatical error identification and recovery:** This can only work with some kind of human intervention, even when setting up a dictionary before processing there is no guarantee that every word will be found or interpreted correctly, so the task can only proceed reliably with someone confirming the suggestions. Medical text has very many words that a standard spell-checker won't recognise, and its abbreviated and idiosyncratic structure doesn't follow grammatical rules, nor should it have to. However, there is a place for correcting words (or recognizing synonyms and abbreviations of words) which has significance for the task at hand – for instance the correction of obvious misspellings of abscess: abcess, absess, abcsess, or abbreviations of it: abs (evaluated in context). Some medical terms cannot safely be automatically corrected – for instance "hypertension" is the opposite of "hypotension" so a misspelling like "hyprtension" cannot be automatically corrected to one or the other. However, in this case it is most likely an error in spelling hypertension since the "r" in the word "hyprtension" suggests that an "e" was dropped whilst typing, whereas dropping an "o" from hypotension would have resulted in "hyptension". A spelling correction system would typically err on the side of caution and offer both hypertension and hypotension options for "hyprtension", whereas it would suggest only hypotension for "hyptension".
- **Named entity recognition (NER)**: identifying and classifying words and phrases ('entities') and categorizing them for example, as symptoms, diagnoses, or procedures. Recognizing clinical named entities is not a trivial task. Amongst the issues that make

clinical NER challenging are: word/phrase order variation, derivation, inflection, synonymy and polysemy.

- **Word sense disambiguation (WSD)**: resolving the ambiguity of a string with multiple meanings.
- **Assertion classification:** classifying status of a medical concept. This task was introduced by Uzuner et al as part of i2b2 challenge (Uzuner et al., 2011, p. 2) and includes categories of 'Present,' 'Absent,' 'Possible,' 'Conditional,' 'Hypothetical' and 'Not associated with the patient'.
- **Relationship extraction:** Identifying relationships between entities in the text. Examples of relation could be 'treats,' 'improves,' 'worsens,' 'causes,' and 'occurs with.'
- **Temporal relationship extraction**: Identifying and distinguishing expressions referring to time, duration and frequency to situate events in the text.

2.6 Medical Terminologies

Medicine has many available resources to understanding its wide ranging subject areas, including terminologies, classification systems, and ontologies. Almost all of the systems described above to some extent make use of these resources.

Medical *terminologies* focus on comprehensively naming clinical terms, *classification systems* create codes from medical terms, and *ontologies* are a formal specification of the terms belonging to a domain, and their interrelationships (Gruber, 1993). In practice 'ontology' is used as a generic term for a variety of knowledge sources (Fung and Bodenreider, 2012).

Standard ontologies and terminologies are required to successfully integrate information, to search for and exchange data, for understanding and analysis of data etc., in summary - for all knowledge-intensive tasks (Ivanović and Budimac, 2014).

Ontologies define a common vocabulary for a domain, including machine-interpretable definitions of domain concepts and relations among them. An ontology can make domain

assumptions explicit; abstract domain knowledge from operational knowledge; share a common understanding of information structure; and enable domain knowledge analysis and reuse (Fung and Bodenreider, 2012).

Systematic development of medical terminologies started in the mid nineteenth century, but with the advent of computerised systems major initiatives have resulted, chief of which are universally adopted standard terminologies like Systematized Nomenclature of Medicine – Clinical Terms (SNOMED-CT) , Medical Subject Headings (MeSH) and the International Classification of Diseases (ICD) ; and the creation of the Unified Medical Language System (UMLS), which is a compendium of the various vocabularies, allowing for translation between them.

2.6.1 ICD

International Code for Diseases (ICD) is a standard used for coding medical concepts such as diagnoses, diseases, symptoms and procedures by different health organizations including hospitals, policy-makers and insurance companies. ICD-10 is the 10th and current revision on ICD and has been in use since 1994 ("WHO | International Classification of Diseases (ICD)," n.d.).

2.6.2 SNOMED-CT

Systematized Nomenclature of Medicine Clinical Terms (SNOMED-CT) is the most recent edition in a series of SNOMED terminologies that is being maintained by International Health Terminology Standards Development Organisation. SNOMED-CT contains concepts with unique meanings and definitions and is arranged in hierarchies. SNOMED CT is represented using three types of component: concepts, descriptions and relationships.("SNOMED CT," n.d.)

2.6.3 UMLS

The Unified Medical Language System (UMLS) was initiated by the National Library of Medicine and contains many controlled vocabularies including ICD-10 and SNOMED-CT and provides cross referencing across all those vocabularies ("UMLS Quick Start Guide," n.d.).

2.7 Discussion and Analysis

In chapter one the problem that this thesis is addressing has been described as one of discovering the most effective technologies to use for extracting neurosurgical department-specific audit codes from the free text of the attending physicians' notes. In chapter two, the main objectives when extracting codified data from free text have been described as *information extraction*, which is a process of retrieving specific targeted information from text or speech as fixed-format unambiguous data; and as *named entity recognition*, which is a process of categorizing these data into entities of interest.

Exploring the various technologies used for these tasks has revealed that the type of text found in the neurosurgical department is amenable to processing via *pattern matching*, which is suited for text that has a consistent structure, where terms are used repetitively, and also where the text is minimal or the language used is terse and lacking the normal language structures that make text capable of more sophisticated syntactic/semantic natural language processing. Additionally, pattern matching suits targeting of text for specific terms, which has been identified as the desired approach for the neurosurgical department's identification of audit codes (see Table 2-1 for examples of matching of specific terms).

When evaluating the ability of various natural language processing applications for dealing with the non-standard ontology and abbreviated language employed in the neurosurgical department, it was found that their default configuration did not fit the neurosurgical department's needs. Their standard ontologies are not used by the department, and they were not able to correctly identify many of the neurological and neurosurgical terms and abbreviations. The application that came closest to the needs of the department was cTAKES, but any of them would have required a lot of adaptation to perform the department's information extraction tasks. It was estimated that extensive consultation with application experts would be required in order to explore their capacities further, an endeavour that was outside the scope of the current research. Conversely, there was benefit in researching a custom approach that could efficiently perform information

extraction via pattern matching using technologies that could be easily adopted by the neurosurgical department.

Analysis of the capacity for machine learning (ML) showed that ML is able to identify meaningful recurring patterns, provided that enough properly prepared training data is made available for a machine to learn from. Although the neurosurgical department has not provided any suitable training data it was decided that there are most likely sufficient examples where the text of a note matches the audit code assigned to an admission, so that training a machine could proceed, albeit with some degree of "noise", and therefore that there was benefit in adding a ML component to the proposed method. This is elaborated in chapter 4 of the thesis.

In conclusion, an optimal system for the neurosurgical department's auditors is one that quickly identifies areas of the text relative to their usefulness for the task of audit code identification, putting aside text that is not useful, and delivering a precise match to audit codes from the remainder. There should be little required from a person using this system apart from confirming the choices made or choosing alternative or extra codes after examining a small number of valid additional codes. After examining the various possibilities discovered during the literature review, the author decided that a custom application using a combination of rules-based and machine learning pattern matching is required, and this research aim is to design and implement a solution that combines the most effective techniques for identifying words that can be matched to audit codes in this specialised area. The design must be lightweight, responsive and easy to interact with, delivering precise matching from the parts of the text that matter, and capable of being understood and improved.

2.8 Summary

Free-text clinical notes are a natural and expressive means of communication, and properly written clinical notes are more comprehensive and can be more accurate than the equivalent codified data, therefore writing notes continues to be the preferred approach for clinicians. However codified data is required for analytical work, so for that reason much research continues

to be devoted to translating free-form text into codified data through the techniques of information extraction.

The process of information extraction has been reviewed, and the use of various information extraction methods. Most of the literature describes experimental designs that were developed in response to NLP challenges, many of them biomedical in orientation, but when clinical in focus, tending to be devoted to the data made available in the challenges – radiology notes, admission notes and discharge summaries, and pathology results.

The clinical text processing systems reviewed, whether generalizable or highly specific, are designed to match to standard medical coding ontologies. In the evaluation of the generalizable systems it has been observed that they often fetch what seems to be an overabundance of code-matched choices with an accompanying computational overhead, indicating that they would need to be tuned to become useful to a narrower domain.

Chapter 3

Research Design

3.1 Introduction

This chapter describes the research approach used to investigate the problem and devise a solution to it. While the work is fundamentally informed by the human (or behavioural) requirements of the organisation for an effective IT solution to the problem of codifying natural language, a methodical approach was taken when designing and describing the IT solution, using principles derived from Design Science Research. Design Science (Hevner et al., 2004) describes the creation of *artefacts*, amongst which are *methods* and *instantiations* of methods used to solve problems, which is the approach taken. The primary objective of the artefact is described as a method that enables the comprehensive and accurate extraction of codified data suitable for the audit process from the neurosurgical department's free text notes, with an ultimate objective for the method to be utilised at data entry time to suggest diagnostic codes, instantiated as a computer programme and report.

3.2 Research Methodology

The research was conducted according to Design Science principles, as described by Hevner et al. (2004). Design Science is described as the process of creating and evaluating IT *artefacts* which are created to address previously unsolved organizational problems.

An artefact may be described as a construct, a model, a method, or an instantiation.

A *construct* provides the language through which problems and solutions can be defined and communicated (Schön, 1991) – if a problem has never been enumerated before then the first contribution that research needs to make is to create a suitable vocabulary for discussing the problem. A *model* uses the language of constructs to describe a problem and the space its possible solution should exist in (Simon, 1996). *Methods* are the processes by which a problem is proposed

to be solved, and an *instantiation* is the realization of methods in software or hardware that can be applied to the problem (Hevner et al., 2004).

The guiding principle for design science is that understanding a problem and its solution comes about by building, describing and applying an artefact, and that this artefact solves the problem.

Artefacts differ from fully-fledged industry created solutions in a couple of important ways – an artefact is created to explore a problem and its solution space in a way that contributes to research and understanding, and an artefact is not a completed application. In industry very often solutions are created to quickly address problems, not necessarily needing to refer to any design principles and not creating any information that can be used to add to others' understanding of the problem that was addressed. That is, not contributing to research. An application developer might immediately act to solve a problem, without generating anything rigorously defined and clearly articulated that would be useful for others coming across the same problem.

Design science must always proceed with clear and articulated thinking – about how to understand and talk about the problem, how to model it, how to solve it, how to communicate the solution. Design science can however be retrofitted to an industry solution – extracting the design principles embodied in the solution and publishing them in terms of the science of the problem domain.

Design Science also differs from pure design because of the creation of an artefact, through which design is given empirical form and by which a solution space is opened and explored. Donald Norman, one of the leaders in human centred design, has said "In the research university there is much thought, little action. In industry, there is much action, little time for thought." (Norman, 1999, pg.2). In Design Science there is much thought, and enough action to test and articulate that thought comprehensively.

Hevner et al. (2004) list seven guidelines for conducting design science research:

Guideline 1 – Design Science requires an innovative, purposeful *artefact.*

Guideline 2 – that is *relevant* for a specified problem domain.

Guideline 3 – Thorough *evaluation* of the artefact is needed to prove its effectiveness.

Guideline 4 – The artefact should be *innovative*, contributing something new – solving a previously unsolved problem or a known problem more effectively or efficiently.

Guideline 5 – *Research rigor* is required in the definition, internal consistency, coherence and presentation of the artefact.

Guideline 6 – The creative effort required and often the artefact itself opens a *search process*, a problem space is opened and a solution enters into it.

Guideline 7 – *Effective communication* of the research in technological and managerial terms.

In terms of the guidelines, the specified problem domain is one of information extraction - identifying audit codes from neurosurgical department free text. A relevant artefact to address this problem will harness computing power to identify the codes and present them to the department's auditors. Following the guidelines, this thesis is devoted to describing the architecture of an effective computer-based method for identifying the neurosurgical department's audit codes, delivered by way of a computer programme and report that combine the incoming free text with identified audit codes – in design science terms the programme and report together are an instantiation of the method.

Table 3-1 Implementation of Research Guidelines by the Method

Guideline	Explanation	Implementation
Guideline 1 Design as an artefact	Design Science requires an innovative, purposeful artefact (a construct, model, method, or instantiation)	An effective computer-based method for identifying the neurosurgical department's audit codes.

Table 3-1 Implementation of Research Guidelines by the Method

Guideline	Explanation	Implementation
Guideline 2 Problem Relevance	The artefact is relevant for a specified problem domain.	The method addressed the particular information extraction challenges of the neurosurgical department.
Guideline 3 Design Evaluation	Thorough evaluation of the artefact is needed to prove its effectiveness.	The method was evaluated using standard evaluation metrics of precision, recall and F-measure; its instantiation by way of a programme and report delivering audit codes - will be evaluated by neurosurgical department auditors.
Guideline 4 Research Contribution	The artefact should be innovative, contributing something new – solving a previously unsolved problem or a known problem more effectively or efficiently.	An innovative approach to combining rule and machine learning-based predictive techniques resulted in an architecture and computer programme and report that are understandable and modifiable by the neurosurgical department, and that will fit into its computing environment and workflow.
Guideline 5 Research Rigor	Research rigor is required in the definition, internal consistency, coherence and presentation of the artefact.	Research rigor was enforced in the definition of the problem; in the literature review; the analysis of the data; the choice of the most appropriate techniques for the problem; in the design of the method and in the software programming techniques used; in the evaluation

Table 3-1 Implementation of Research Guidelines by the Method

Guideline	Explanation	Implementation
		process and in the presentation of the results in this thesis.
Guideline 6 Design as a Search process	The creative effort required and often the artefact itself opens a search process, a problem space is opened and a solution enters into it.	The architecture of the method allows for iterative experimentation in the search for the most effective solution. The research process opened up new possibilities for effectively using intelligent technologies.
Guideline 7 Communication of Research	Effective communication of the research in technological and managerial terms.	The thesis was presented using terminology that explains the technology utilised, but with sufficient explanation for the research to be understood by any reader.

The next section describes the problem domain in detail.

3.3 Problem Domain

The Neurosurgical Department has an application that is used to describe the diagnoses and procedures performed for each patient passing through the department. This is independent of the main electronic records system of the hospital, the data collected and terms used are highly specific to neurological conditions and neurosurgical procedures.

In the hospital, initial notes taken about a patient's condition and diagnoses are recorded on paper, using whatever note-taking system the interviewing Neurosurgical Department registrar prefers. If a patient is then admitted to the department these notes are transcribed into the department's "red book" by the registrar or a resident, and other notes are added. By the end of the day the "red

book" information is entered by a resident into the Neurosurgical Department patient management system. Consequently, there are at least two people and potentially three transcriptions between the initial interview and the data being entered to the system.

Any further information during patient journey including procedures will also be added to the patient record. The information then gets printed and reviewed in a weekly audit meeting to verify correctness of information from a very high level.

The ontology that is being used in the system is a customized specific ontology devised by one of the hospital registrars that is close to the vocabulary used in the "red book", it has been subsequently revised in 2008 to have a better coverage of diagnosis and procedures.

The structure of a record in the application is one of a code and an accompanying note, with as many records per admission as is required to properly code all of the diagnoses and procedures. A note is not required against a code, though it is expected that a note will appropriately qualify and amplify the code picked.

The codes are then used for analysis in an audit process, which is looking for certain codes of interest, especially those that pertain to measuring the performance of the neurosurgical department. For example, data that describes correct procedures in relation to diagnoses, and data about infections and other complications arising from procedures.

The problem that the audit process faces is that codes may be inappropriately or insufficiently picked, so that there is not enough data to properly audit an admission. Therefore, codes need to be fixed or added in order to complete the data, which takes place sometime after the patient has been discharged - in preparation for the annual auditing cycle.

Where notes have been used it is possible for a domain expert to identify the appropriate codes that pertain, and to re-code or add new codes - but this is demanding and requires more resources than the department has available. The desired improvement to this situation is to use computing

technology to identify when incorrect or insufficient codes have been used, and to suggest the correct codes. The next section describes the method by which the problem is addressed, adhering to the guidelines set forth above.

3.4 Method Design

The initial objective of the method is to enable comprehensive and accurate extraction of codified data suitable for the audit process from the neurosurgical department's records. The ultimate objective is for the method to be utilised at initial data entry to suggest appropriate diagnostic codes, which are more detailed and varied than the auditing codes. The artefact is a computer-based method designed to generate audit code suggestions based on text classification from notes attached to a record, delivered via a computer programme and report, which will enhance the process of reviewing the notes by a system expert at the time of auditing. The general architecture of the method and its workflow are shown in Figure 3.1.

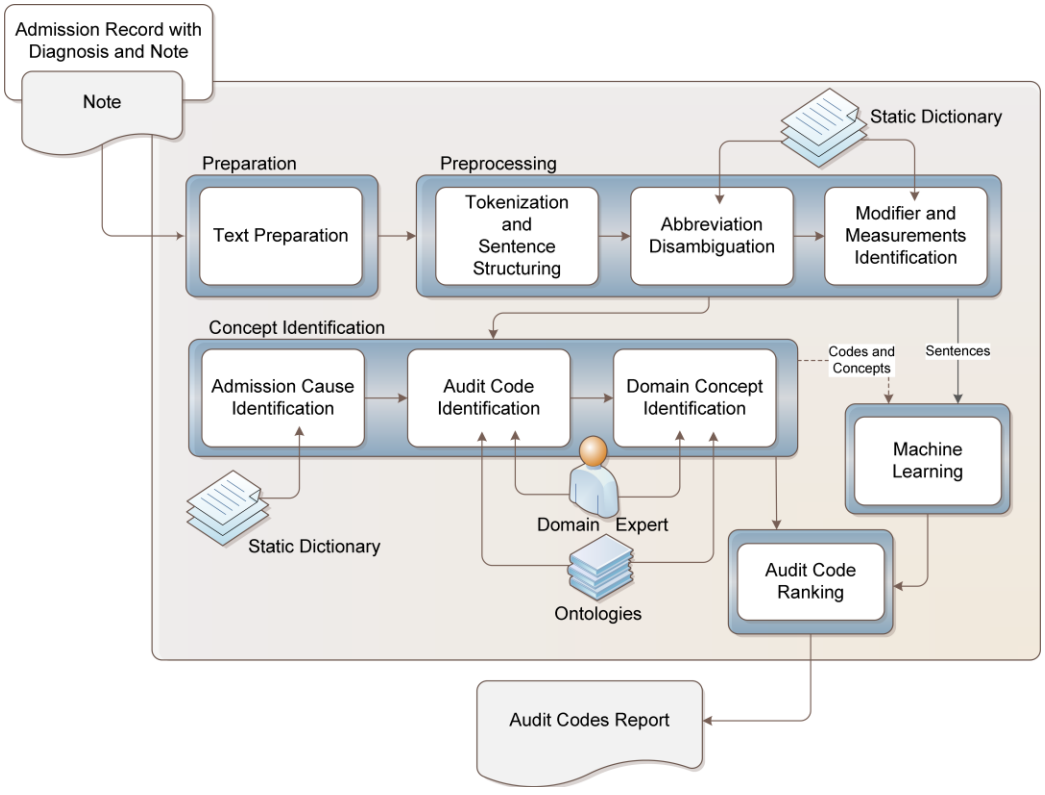


Figure 3.1 Architecture of the audit code extraction method

To do this the structure and language of the Note is first regularised, then broken it into its constituent parts, and finally those parts are identified as belonging to either audit codes or to other useful concepts (or domains). These classifications may be done on an individual word basis but also take into account surrounding words. The process of determining audit codes and other concepts is broken into five stages – Preparation, Pre-processing, rules-based Concept Identification, Machine Learning, with a final step of Audit Code Ranking.

The audit code extraction method consists of three initial consecutive stages, with the output of a stage providing the input for the next stage: Preparation of the text, followed by Pre-Processing, then by Concept Identification. One of the outputs of pre-processing are formatted sentences, which are handed to the Machine Learning stage for its prediction process. It is envisaged that the Concept Identification outputs of identified audit codes and domain concepts can also be imported into the machine learning algorithm, so that it is able to utilise the additional meta-data. The outputs of the Concept Identification and Machine Learning are audit codes, which are finally ranked in terms of their frequency and consistency with the record's diagnosis, and saved back into the data. Finally, a report of audit codes is created for the department's auditors.

Preparation ensures that the text is ready for the various steps in pre-processing, especially tokenization which requires that words are properly separated from one another. Pre-processing consists of all the steps that break the text into words and sentences, and there is also some identification of words performed, based on static dictionaries of words. Concept Identification is where useful words and phrases are identified – the outcome of this is that some words and phrases are tagged as audit codes, others as belonging to various domains. Each step here utilises either a static dictionary of terms, or an ontology – meaning a dictionary of medical terms that is continuously being refined with feedback from a domain expert after analysis of the outputs of concept identification. A rules-based computer programme is throughout the initial 3 stages, but the Machine Learning process uses the classification method obtained by training a machine.

3.4.1 Preparation Step

The Preparation step includes:

1) Setting up the environment including dictionaries
2) Fixing text boundaries prior to tokenization
3) Some Abbreviation Disambiguation
4) Spelling correction and regularization of key words

Preparation of the Notes is required before the pre-processing stage, so that pre-processing works correctly. One of the tasks of pre-processing is to identify and split words by finding a space between words, and to split the Note into sentences by finding sentence boundary characters such as a full stop. Therefore, the Note requires processing to enforce spaces between words, and around the sentence boundary characters, so that they are correctly identified. At the same spaces and sentence boundary characters need to be eliminated where they should not exist, so that words and sentences are not split erroneously.

The Preparation Stage ensures that every significant word or character is surrounded with a space, including sentence boundary characters. Certain words or patterns of words are dealt with to ensure splitting of them does not occur.

A dictionary is used to fix any commonly occurring spelling mistakes of important words at this point, as the Notes to be consistent as far as possible when using these words, which are the basis of the majority of the diagnoses classifications. Also at this point various ambiguous abbreviations are translated into precise and consistent terms, so that at any later stages of abbreviation translation these are not dealt with incorrectly. The list of dictionaries used for pre-processing and concept identification is shown in Table 3-2.

Table 3-2 Dictionaries used in Pre-Processing and Concept Identification

Category	**Example**
Stop characters	. ; , : ! - -- + and
Punctuation	( ) [ ] * ^ % $ @ / \ _ " < ' >
Negations	no, not, non, none, nil, -ve, negative, exclude, NAD, etc.
Positions	(L, L), R, R), left, right, adjacent, upper, lower, posterior etc.
Probabilities	probable, ? , suspected, unknown, investigation etc.
Actions	-->, ->, for, await, awaiting, withdrawn etc.
Severities	mod, moderate, high, medium, low, severe, ++, +++, +ve, ~, etc.
Noise words	derived from SQL standard noise list, with certain symbols like #, numbers, and single letters removed (i.e. not considered noise)
Admission Cause	ped, pedestrian, head, driver, passenger, swimmer, mva, car, truck etc.
Admission Cause 2	fall, fell, collapse, assaulted, ETOH, intoxicated, bed, scaffold etc.
Domain Concepts	Definitions of anatomical, diagnostic, procedural and other words

Preparation Step examples:

1) Elimination of unwanted sentence boundary characters can be seen in measurements that include a full stop, here the process substitutes double pipe characters (e.g. 1.5 mm → 1||5 mm) until tokenization is completed, after which the pipe symbols are replaced with a full stop to restore the measurement.
2) An example of expanding ambiguous abbreviations can be found in the abbreviation “art”: Replace “fem art” with “femoral artery”, “sup art” with “superior articular”. Normalizing and correcting important terms into a single meaningful word or phrase: “Nsurg”, “neurosurg”, “nurosugical”, “N/S” all get translated to “neurosurgical”.

3.4.2 Preprocessing Step

Preprocessing steps include:

1) Tokenization and Sentence Structuring:
2) Lexical Tagging
3) Abbreviation Disambiguation

The Preprocessing stage extracts every word from the Notes, and groups them in sentences with numeric identifiers to enable word relationship processing. The primary purpose of the next step, Concept Identification, is to identify which single or multi-word phrases are relevant to reaching a diagnosis, but this step is performing a preliminary identification of the words during lexical tagging, to label those which are noise, punctuation, sentence boundaries, and a number of other types on the way to discovering what words remain as key words per sentence.

The additional types of words identified during lexical tagging include negations, probabilities, positional words, severity, actions, dates, time, tests and measurements. It is more efficient to identify these now since the type of processing they require is essentially identical to that used to identify noise and punctuation – comparison against either a fixed set of words or a limited set of patterns.

3.4.3 Concept Identification Step

Concept Identification steps include:

1) Admission Cause Identification
2) Diagnostic Concept Identification
3) Domain Concept Identification

After the Preprocessing words are collected into groups within sentences, ready for further processing. Some of the words have already been identified, those which remain are therefore potentially meaningful for labelling for diagnosis, and the labelling is the task of the Concept Identification Stage.

Admission Cause Identification is performed first, as there is value in identifying as much as possible of the Note that does not directly relate to the identification of a diagnosis, so that there are fewer remaining words for the Audit Code identification stage to analyse. It is also possible that an understanding of the cause of admission could help in identifying an audit code. Admission Cause can often be identified by a phrase such as “ped vs car”, meaning a pedestrian struck by a

car, or by “ETOH, fall” meaning falling over while intoxicated (ETOH is an abbreviation for alcohol). Two overlapping static dictionaries of terms are employed to allow these sorts of phrases to be identified.

Audit Code identification is a routine that examines the words one at a time, to see if they can be ascribed to an audit code, based on rules established by a system expert. If necessary, the word being dealt with will be compared to other words in its containing sentence or even to all of the key words found in the Note. A summary of the main audit codes is shown in Table 3-3.

Table 3-3 Audit Codes used in the neurosurgery department

Concept (Cranial)	**Concept (Spine)**	**Concept (Neoplasia and Lesion)**
ANEURYSM	SPINE:TRAUMA	CRANIAL:NEOPLASIA
AVM	SPINE:TRAUMA:FRACTURE	CRANIAL:NEOPLASIA:CYST
CSF:LEAK	SPINE:TRAUMA:CORD	CRANIAL:NEOPLASIA:GLIOMA
CRANIAL:TRAUMA	SPINE:TRAUMA:DISCO-LIGAMENTOUS	CRANIAL:NEOPLASIA:MENINGIOMA
CRANIAL:TRAUMA:SKULL FRACTURE	SPINE:CANAL STENOSIS	CRANIAL:NEOPLASIA:METASTASIS
CRANIAL:TRAUMA:CONTUSIONS	SPINE:CAVERNOMA	CRANIAL:NEOPLASIA:PITUITARY
CRANIAL:TRAUMA:EDH	SPINE:DEGENERATIVE	CRANIAL:NEOPLASIA:SCHWANNOMA
CRANIAL:TRAUMA:ICH	SPINE:OTHER	CRANIAL:CAVERNOMA
CRANIAL:TRAUMA:IVH	OTHER:FRACTURE	SPINE:NEOPLASIA
CRANIAL:TRAUMA:SAH	OTHER	FISTULA
CRANIAL:TRAUMA:SDH		LESION
CRANIAL:TRAUMA:TBI		COMPLICATION:INFECTION
HYDROCEPHALUS		

The rules are continually refined by discovering additional rules based on conclusions reached after manual inspection of the processed Notes - examining which words are commonly used with a given diagnosis. The rule refinements require confirmation by a system expert, who developed the original rules as wildcard searches within the Notes – for instance in plain English: “show me all Notes containing the word ‘SDH’ or with words like ‘subdural’ and ‘haem’ (as in haematoma or haemorrhage)”. The design of the rules used here reflects that approach, but with the emphasis on identifying all available audit codes per Note, rather than multiple retrievals of all Notes per audit code, though that is also possible. Having the ability to identify words and sentences individually allows us to more easily identify audit codes as belonging to a word or more likely a

group of words in a sentence, and thus to match that part of the Note with an audit code, leaving the remaining parts of the Note to also be matched to other codes where possible.

Domain Concept Identification is a third part of Concept Identification process - it identifies any remaining unidentified words by passing them through a dictionary, so that as an ideal final outcome all words of significance will be identified. Those that are useful for deducing diagnoses will become the basis for introducing the words to the any future stage of making the diagnoses; and together with the audit codes they can also be passed into the machine learning stage as meta-data for the sentences it is processing.

Table 3-4 shows the summary of Domain Concepts.

Table 3-4 Domain Concepts

Concept (anatomical)	**Concept (diagnostic)**	**Concept (procedural)**	**Concept (Other)**
anatomical:brain	diagnostic:brain	procedural:brain	location:geographic
anatomical:brain,spine	diagnostic:heart	procedural:heart	medication:vascular
anatomical:skull	diagnostic:infection	procedural:skull	
anatomical:skull,spine	diagnostic:neoplasm	procedural:spine	
anatomical:spine	diagnostic:neurological	procedural	
anatomical:vascular	diagnostic:vascular	test:brain	
anatomical	diagnostic	test:spine	
	history	test	

The rules used are firstly based on the patterns supplied by the auditing department system expert, and so they are orientated around the requirements of clinical auditing, and therefore sometimes they are not precise in their reach compared to the requirement of a specific diagnosis. Rules are expected to be refined to align them more with the requirements of specific diagnosis, and the dictionary will be continuously added to.

3.4.4 Machine Learning (ML) Component

Individual sentences and additionally the entire Note were presented to various machine learning algorithms in order to evaluate them, and the Weka (Witten and Frank, 2005) Sequential Minimal Optimization (SMO) Support Vector Machine (SVM) was found to be the most accurate (see Table 3-5) and to provide a useful output that can be incorporated into the method. The Weka SMO Support Vector Machine was trained using sentences classified as containing only one audit code, with the result being a model that classifies audit codes by comparing the score of the text associated with a code in a series of binary tests – one per code vs every other code.

A machine learning algorithm can then be used to predict new text, by passing into the ML software a similarly prepared but new batch of data that was used to train the algorithm. As this batch process is an undesirable overhead and difficult to incorporate into a lightweight and efficient method a different approach was taken: the ML model's weighted word attributes per class were extracted from the SVM model as a table of weighted words per audit code, which was then incorporated into the method's machine learning component. Therefore, although the ML training itself is a separate task, the use of the ML attribute data is incorporated seamlessly into the method, meaning the classification of new text takes place within the method.

To do this, every word in each sentence is related to the SVM attribute weights table, and the weights per sentence are summed. The audit code of the group that delivers the highest attribute weight score is chosen as the code of the sentence. This information is combined with the output of the rule-based Concept Identification stage to supplement the data found there, with the final step of audit code ranking taking care of prioritising the inputs and filtering out false positives. SVM attribute weights have been used in a similar way to assist in cancer gene feature selection by Guyon et al. (Guyon et al., 2002).

There is scope to further train machine learning algorithms by including the classified words coming from the Concept Identification process, but it is expected that any further ML models would also be incorporated seamlessly into the method, so that predictions are online, and could be included into a future application that is used real-time during data entry.

3.4.5 Audit Code Ranking Component

The audit codes delivered by the rules-based Concept Identification process are the primary source, but if that process has failed to find a code and the machine learning process has delivered one, or has found additional relevant codes, then they are added to the list of reportable codes.

Although the accuracy of the method used must be assessed by comparing the audit code which is predicted from the Notes against the audit code of the admission record, the immediate purpose of the process is to find any extra codes that have not been attached to the admission. To this end the audit codes that are discovered are compared to the audit code of the admission record, and between themselves for consistency and precision. When the admission audit code is identified by the model, then that indicates the data entry person picked a diagnosis that is consistent with the Note, and so the model can rely on this as an example; therefore, further codes of a similar nature can also be confidently presented.

Admission audit codes that are interchangeable with the one chosen are differentiated from those that are definitely additional - for instance the code CRANIAL:TRAUMA:TBI is often interchangeable with CRANIAL:TRAUMA:CONTUSIONS, so reporting the alternative is not as useful as reporting codes that are definitely needing to be added to the main code, such as CRANIAL:TRAUMA:SDH.

The audit codes found are sometimes qualified by surrounding words indicating uncertainty, or negations. These must be taken into account when ranking the codes, so that those with the highest chance of being correct to be reported on are presented first. The codes most clearly like the audit code of the admission record are ranked higher than those that would seem to be irrelevant, with those that seem clearly out of step being noted as unlikely.

In summary, the Diagnostic Concept rule-based step combined with the Machine Learning step delivers words identified as belonging to an Audit Code. The Domain Concept dictionary step labels any other word with information about what domain it belongs to, which can also be fed into a Machine Learning algorithm, and which may assist in some later diagnosis step.

3.5 Machine Learning Models Evaluation

In order to implement the machine learning component, a trial was conducted of various supervised machine learning techniques, using data prepared from the neurosurgical department (see sections 4.4 and 4.5). The three which were found to be most accurate and useful have been widely used in this domain: Naïve Bayes (NB), Decision Trees, and SMO - an implementation of a support vector machine (SVM) algorithm.

Table 3-5 Comparisons of Machine Learning methods' correct classification

Percentage records	J48	NBM	SVM
correctly classified	59.8%	62.0%	63.1%

Naïve Bayes Classifiers (Lewis, 1998) are based on probabilistic theory and assume that all attributes in the context of each class are independent of each other and has successfully been employed in similar situations (Fiszman et al., 2000; Luke Butt, 2013; Pakhomov et al., 2006; Zuccon et al., 2013). The Multinominal variation of Naïve Bayes performed marginally better than standard Naïve Bayes, as it also ranks words based on their counts.

Decision trees are constructed through iteratively selecting a set of attributes that splits the data into subsets. To build the tree a mathematical algorithm is used to find a variable and its threshold that divides the data into subsets (Leo. Breiman, 1993; Quinlan, 1993). A J48 decision tree was used, which is Weka implementation of C4.5. Decision trees have proved to be effective in classification tasks of clinical domain (Farkas and Szarvas, 2008; Huang et al., 2007; Spat et al., 2007).

Support Vector Machines (SVMs) (Cortes and Vapnik, 1995) is a powerful learning method and is successfully applied to other similar tasks (Aronson et al., 2007; Clark et al., 2008). The SMO implementation of SVM was found to be most useful, it uses the sequential minimal optimization algorithm (Platt, 1999) for training a linear support vector machine.

The SVM was found to be the most useful model: it was the most accurate, reasonably quick, and produced a useful output that could be incorporated back into the audit classifier method. The SVM compares each class with every other class in turn, a binary class comparison that determines which words most successfully differentiate each class from every other class, and scores them. The words and their weighted scores per associated class comparison are preserved in the text output of the ML process. These scored class comparison words are then imported as a table into the underlying database of the audit classifier method, and form the basis for its machine learning-based prediction algorithm. The audit code classifier's ML algorithm adds the combined weight of the words in a sentence by relating them to the table of SVM weighted words, to discover the strongest scoring class of all the binary comparisons that use these words - this becomes the ML prediction. Essentially the output of the ML model is data that allows the audit classifier to then reverse engineer the predicted ML class, bypassing the need to feed sentences back to the original ML software for a prediction.

All the machine learning algorithms were implemented using Weka data mining software version 3.7.13 (Witten and Frank, 2005) .

3.6 Conclusion

This chapter described the use of Design Science as a methodology for designing the architecture of a method to solve the problem of extracting audit codes from the free text notes of the neurosurgical department of a major trauma hospital. The method is to be instantiated as a verifiable computer programme and report. The method uses natural language preprocessing to identify important sentences and words, and pattern matching techniques informed by a system expert, combined with a machine learning model, to match these sentences to audit codes. The programme is to be constructed using SQL Server programming code and the report written using SQL Server Reporting Services, so that these might be evaluated in place in the neurosurgical department's computer systems, this implementation is discussed in detail in the next chapter.

Chapter 4
Research Results and Evaluation

4.1 Introduction

The previous chapter described the audit code extraction method, including the key rule-based and machine-learning techniques used by the method. In this chapter the evaluation of these techniques is described, assisted at times by a more technical discussion of the techniques. The chapter covers techniques used to combine rule and machine learning systems, and details the calculation of the additional and alternative audit codes that are the purpose of the method. Various challenges encountered during the evaluation are described, and the chapter concludes with a presentation of a report of the predicted additional and alternative audit codes.

4.2 Evaluation Aims and Objectives

The evaluation process aims to analyse the success of the audit code extraction process in deriving audit codes from free text, by using standard measures that describe how accurate a system is. As both rule-based and machine learning methods are being used the evaluation must compare the two approaches but also help to identify the best combination of the two. The objective is the construction of a hybrid system that combines the strengths of rule-based and machine learning classification methods, being both precise and with good coverage.

Although a correct match to the incoming audit code is a desirable feature of the classifier it is not the final objective, as the system is designed to discover relevant additional codes, or to suggest more accurate codes in the case of the incoming code being an imprecise one. Discovering text that proves the incoming code is however the primary way that accuracy can be measured, as the incoming code is the de facto standard to measure against. If the system matches these codes with a high degree of accuracy then there can be confidence that it is also finding the additional codes with a similar accuracy, furthermore the words that suggest the additional codes

can be cross checked against records where these codes constitute the primary, incoming audit code.

4.3 Evaluation Measures

The research employs the standard evaluation metrics of Precision, Recall and F-score that clinical narrative classification typically uses to evaluate the success of an NLP system (Stanfill et al., 2010). The values for Precision, Recall and F-score are expressed as a decimal fraction of 1, or as percentages.

Precision, Recall, and F Measure are formulae to describe the relationships between the correct codes (or classes) and the tool's ability to identify them. To describe how a tool identifies classes the idea of positive and negative class identification is used – a positive when a class is identified as being present, and a negative when it is identified as not being present. A class may be correctly identified – a true identification, or incorrectly identified – a false identification. Therefore, there are four possible results of the identification process:
True Positive (TP) – a class is correctly identified as being present
True Negative (TN) – a class is correctly identified as being absent
False Positive (FP) - a class is incorrectly identified as being present
False Negative (FN) - a class is incorrectly identified as being absent

Precision (P) is defined as the ratio of correctly suggested classes (true positives) to the total number of classes *suggested* by the identification method (true positives plus false positives):
Precision (P) = TP / (TP + FP)

Recall (R) is the ratio of correctly suggested audit classes (true positives) to the number of *actual* classes in the data (true positives and false negatives).
Recall (R) = TP / (TP + FN)

The two measures of recall and precision trade off against one another, therefore an increase in precision will usually result in a decrease in recall and vice versa. F-measure is a single measure that combines precision and recall and is defined as the harmonic mean of precision and recall (Christopher D. Manning, 1999).

$F = 2 \times P \times R / (P + R)$. Alternatively: $1 / ((0.5 / P) + (0.5 / R))$

The values for recall, precision, and F-measure will be between 0-1 with 1 being the optimum result, which for ease of understanding are often expressed as percentages.

Two methods for calculating overall evaluation measures have been used: micro-averaging and macro-averaging, as explained by Manning (Christopher D. Manning, 1999). Micro-averaging computes the summation of all the individual class scores into one contingency table and then the evaluation measures are calculated based on the totals in this table. Macro-averaging calculates recall, precision, and F-measure for each class, and then computes weighted averages of these scores. Micro-averaging gives more importance to the classes with larger number of instances as their influence predominates, whereas macro-averaging evaluates the performance of the classifier across all the classes more fairly (Christopher D. Manning, 1999; Jackson, 2002). See Appendix D for an example of the different approaches.

Micro-averaging is typically used in systems where classes containing the largest number of records are the most important, for instance applications that need to verify that the majority of medical cases are correctly billed, rather than needing to precisely discover every diagnostic code (Jackson and Moulinier, 2002). As macro-averaging more accurately estimates the consistency of a classification system it is the preferred approach here (though both are calculated), and in fact scores better than a micro-averaging approach for the neurosurgical audit code classifications.

4.4 Dataset

The dataset spans ten years from mid-2003 to the end of 2014; the diagnostic data is contained in a table of 24,437 records, and the procedural data table has 11,721 records. The diagnostic codes have six levels of hierarchy with 6 root nodes and a total of 213 diagnoses. The 6 root nodes include such codes as "Spinal" "Cranial" or "Complications." Procedural codes have only two levels of hierarchy but 12 root nodes with a total of 134 procedures. Each diagnosis is associated with an audit code, which is very often a higher level classification that encompasses a number of diagnoses, but is sometimes as specific as the diagnosis; examples are given in Table 4-1.

Table 4-1 Diagnosis Code and Audit Code structure

Diagnosis Code	Diagnosis Value	Audit Code
218-224	Cranial>Trauma	CRANIAL:TRAUMA
218-224-309	Cranial>Trauma>Osseous Injury	CRANIAL:TRAUMA:SKULL FRACTURE
218-224-309-310	Cranial>Trauma>Osseous Injury>Skull	CRANIAL:TRAUMA:SKULL FRACTURE
218-224-309-310-314	Cranial>Trauma>Osseous Injury>Skull>Non-displaced	CRANIAL:TRAUMA:SKULL FRACTURE
218-224-309-310-315	Cranial>Trauma>Osseous Injury>Skull>Depressed	CRANIAL:TRAUMA:SKULL FRACTURE
218-224-309-310-315-316	Cranial>Trauma>Osseous Injury>Skull>Depressed>Open	CRANIAL:TRAUMA:SKULL FRACTURE
218-224-309-311	Cranial>Trauma>Osseous Injury>Sinuses	CRANIAL:TRAUMA:SKULL FRACTURE
218-224-309-312	Cranial>Trauma>Osseous Injury>Base of Skull	CRANIAL:TRAUMA:SKULL FRACTURE
218-224-309-350	Cranial>Trauma>Osseous Injury>Facial	CRANIAL:TRAUMA:SKULL FRACTURE
218-220-251	Cranial>Neoplasia>Extrinsic	CRANIAL:NEOPLASIA
218-220-251-243	Cranial>Neoplasia>Extrinsic>Acoustic Neuroma	CRANIAL:NEOPLASIA
218-220-251-242	Cranial>Neoplasia>Extrinsic>Meningioma	CRANIAL:NEOPLASIA:MENINGIOMA
218-220-251-244	Cranial>Neoplasia>Extrinsic>Pituitary	CRANIAL:NEOPLASIA:PITUITARY

4.5 Training and Testing Data

It is standard practice when creating systems that are designed to make predictions from data to train the systems on a separate set of data from the data used to evaluate their accuracy (Bellazzi and Zupan, 2008). These are known as training and testing data. The process of developing a machine learning system using training data is normally described as "learning" the system. Ideally the training data should be a very large collection of as many examples as are required for the system to be able to statistically rank differences between the language patterns used by each

class, and the test data should also have sufficient examples of similar patterns to test if the trained system has correctly identified each class. To properly do this the data should be assembled by system experts from original but de-identified records, into a "gold-standard" data set that comprehensively and accurately represents the original data. If gold-standard quality data cannot be obtained, then ideally there should at least be sufficient numbers of examples for a statistical pattern to emerge - which has the effect of excluding any infrequently encountered invalid data as statistically insignificant.

If a training/test approach is not taken and a machine learning system is learned on all available data, then the resulting system is invalid: on the one hand if it has been learned on all of the data then it is exactly matched to it – a condition called overfitting; and on the other hand there is no data left to validate its accuracy. An over-fitted system shows great accuracy on the data it has been learned on, but is too specific and so is not as capable as a less specifically trained system of recognising statistical patterns in new data. It is not until a system is tested on a good collection of new data that its true accuracy can be measured, so it always the case that some data needs to be reserved for testing.

When learning a system, the training data itself should then be split into training and testing components so that the system can measure its prediction performance as it builds its model; a typical approach is to get the system to do this itself using a process called cross-validation (Witten et al., 2011). When doing cross-validation the machine-learning process repeatedly splits the training data into training and test sets, and consecutively learns and tests itself on each split – typically for 10 iterations, which is called 10-fold cross-validation, abbreviated as 10FCV.

4.6 Machine Learning Data Preparation

The data provided by the neurosurgical department had not been supplied as gold-standard training and test sets, and so the approach taken was to use a random record-selection algorithm to split the data 90/10 into training and test data. However, as some audit codes were represented by very few records it was decided that there should be a minimum of 10 records required before

some could be split off for testing, consequently some audit codes were represented only in the training data, and the final ratio was 90.2 to 9.8 percent, with the data consisting of 12,104 training records covering 66 audit codes, and 1,314 testing records of 53 audit codes.

A 90/10 ratio was picked rather than a more conventional 70/30 due to a number of factors:

a) The 10-fold cross validation process itself divides data 10 times into training and testing data using a 70/30 split, and averages the results. This does not require a separate test set, but we used one as a comparative evaluation approach for the sake of completeness.
b) Because some test data was retained, but there was a lack of data for some audit codes, it was preferred to favour the training data by taking the higher ratio.
c) The eventual purpose of the SVM machine is to detect additional codes by incorporating its predictive power into the model, so while it was important to establish how well it did during the learning process it was more important that it had the best opportunity to learn on a variety a records. This was also necessary due to the rather arbitrary nature of much of the notes data in relation to the incoming audit code, a greater number of records meant a higher number of examples that were actually correct for the audit code.
d) Testing with a test set was tried at a 70/30 split and the result did not differ markedly from that obtained using a 90/10 split. In both cases the figures obtained by testing the model against the test data were much higher than those obtained when using cross validation, which was a more thorough approach and was therefore preferred. Subsequently when we incorporated the machine learning into our architecture, the figures were close to those obtained by the cross validation.

As an example of the data available to the machine learning when using a 70/30 split vs a 90/10 one. Table 4-2 below shows some of the numbers for the first 25 records and last 5 records, when ordered by available records for training. Table 4-3 illustrates the differences between using 10-fold cross validation and evaluating a test set, when the data is split 70/30 vs 90/10. The last column of the table additionally shows the results of performing a 10-fold cross validation on all of the data.

Table 4-2 Records available using a 70/30 vs 90/10 split

Audit Code	At 70-30		At 90-10	
	Testing	Training	Testing	Training
FUNCTIONAL DISORDER:SPASTICITY		1		1
FUNCTIONAL DISORDER:DYSTONIA		2		2
FUNCTIONAL DISORDER:TREMOR		2		2
SPINE:CAVERNOMA		2		2
SYRINX		3		3
SPINE:AVM		3		3
FUNCTIONAL DISORDER:PARKINSONS		4		4
FUNCTIONAL DISORDER		4		4
SPINE:DAVF		5		5
PERIPHERAL NERVE:NEOPLASIA	1	7		8
PERIPHERAL NERVE:TRAUMA	1	7		8
CHIARI	1	7		8
CRANIAL:NEOPLASIA:CYST	2	7		9
COMPLICATION:DEVICE	3	10	1	12
SPINE:VASCULAR	3	12	1	14
SPINE:SYRINX	3	12	1	14
CRANIAL:CAVERNOMA	3	13	1	15
SPINE:CSF DISORDER	3	14	1	16
SPINE:CSF LEAK	5	14	1	18
CRANIAL:FISTULA	6	16	2	20
PERIPHERAL NERVE	6	21	2	25
ULNAR NERVE	9	23	3	29
COMPLICATION:NEUROLOGICAL	9	25	3	31
CRANIAL:TRAUMA:IVH	9	28	3	34
CRANIAL:CSF LEAK	11	28	3	34
.....				
COMPLICATION	168	397	56	509
SPINE:DEGENERATIVE	222	520	74	668
CRANIAL:TRAUMA:TBI	267	627	89	805
CRANIAL:TRAUMA:SDH	411	964	137	1238
SPINE:TRAUMA:FRACTURE	588	1372	196	1764

Table 4-3 Evaluation differences between a 70/30 and 90/10 split

	70-30		90-10		
	Training 10FCV	Testing	Training 10FCV	Testing	All Data 10FCV
Precision	0.600	0.730	0.611	0.768	0.648
Recall	0.620	0.728	0.631	0.759	0.664
F-measure	0.601	0.718	0.612	0.749	0.645

The paucity of some data had repercussions in the machine learning process and in evaluation, which are described later. There are two record types which the rule-based system is designed to *not* predict – those which are classified simply as "Other", and those that have not been properly classified – which is to say those given the Diagnosis of "Cranial" with no further detail – there is no corresponding audit code for these. Since the rule-based system tries to find more meaningful codes than "Other" and "Unclassified" and doesn't try to predict these (though it may use them as a default last-resort classification), in order to derive a fair comparison between the approaches none of these records were presented to the machine learning process during the initial training and testing evaluations.

The data was converted into Weka machine learning ARFF (Attribute-Relation File Format) files, using the Weka software to process CSV files that had been exported from the original SQL database. In order to make the training and test data sets compatible they needed to be converted together, so that the words and classes (audit codes) were in common across both data sets, though of course not necessarily to be found in both data sets. A number of experiments were performed to understand the best performing data, the final approach used was to use a String to Word Vector filter, with lower case conversion of words, term frequency–inverse document frequency (TF-IDF) transformation, with word counts output. There was no stemming performed, and no stop words list was used in the ML data preparation, however the sentences submitted were pre-processed by the rules-based system - which had eliminated stop words. The resulting prepared data is a "bag-of-words" per sentence with each word assigned a numerical score based on its frequency in the sentence and the inverse of its frequency in the entire collection.

TF-IDF transformation gives a lower score to words which appear frequently over the entire collection, so that they do not assume the same significance as more rarely encountered words, but within each document (in this case each sentence) words are then additionally scored based on their frequency (Turney and Pantel, 2010). The bag-of-words approach to processing free text takes no account of sentence structure or lexical meaning of words, they are simply evaluated based on their scored frequency. The rules-based system however, scores words based on rules about their significance – it looks for specific words as having a positive value, and assigns an

even higher value to some of these based on their relationship to the incoming audit code. Other words are retained without scoring them, though they may serve a function to adjust the scores of the identified words – the rules-based approach takes a rudimentary account of negations and qualifications when scoring words. Like the ML system, the rules-based system additionally scores identified words based on their frequency within a document.

The individual pre-processed sentences of an admissions note (rather than the complete note) were paired with the audit code of the admission, consequently in some instances a sentence did not match the admission's audit code. In most cases these sorts of sentences would be elsewhere correctly paired to their admission audit code - where they were the primary sentence of an admission record.

4.7 Machine Learning Algorithm Selection

Various ML algorithms were assessed, using 10-fold cross validation on the training data, and the three most effective were J48 Decision Tree (J48) (59.8% recall on the training data), Multinominal Naïve Bayes (NMB) (62.0% recall), and Sequential Minimal Optimization (SMO) Support Vector Machine (63.1% recall). Each method was also tested against the test set, producing recall accuracies of 69.5% for J48, 71.2% for NMB, and 75.9% for SVM. The improvement of results on the test sets seems to be too great, some of it may be attributed to the much smaller, and for some classes missing, data set – there was less variation in the data compared with the training set. When the ML systems were learned against all of the data then they performed a few percent better than with just the training data, but not nearly as well as they did with the test data, which also indicates that the test data is not properly representative, and underscores a conclusion that the lack of properly prepared gold-standard training and test data limits the ability to properly evaluate the ML approach. Table 4-4 summarises these results.

Table 4-4 Evaluation Measures of Machine Learning systems

	J48			NBM			SVM		
	Training 10FCV	Testing	All Data 10FCV	Training 10FCV	Testing	All Data 10FCV	Training 10FCV	Testing	All Data 10FCV
Precision	0.579	0.691	0.592	0.609	0.716	0.634	0.611	0.768	0.648
Recall	0.598	0.695	0.613	0.620	0.712	0.647	0.631	0.759	0.664
F-measure	0.577	0.679	0.590	0.608	0.705	0.634	0.612	0.749	0.645

4.8 Evaluation of Predicted Audit Codes

The objective of the neurosurgical department's audit code extraction system is to discover *additional* audit codes after verification of the codes assigned to an admission. Therefore, although it is important to identify the text that delivers an audit code that matches the admission record's audit code, this audit code will not be used beyond the verification purpose – the system's focus on reporting additional audit codes means that it is not useful to report the already identified audit code. Establishing the degree of prediction accuracy is however useful for identifying the classes where a classification approach performs best, as there can then be confidence in using predictions for these audit codes, even when they are not in agreement with the currently designated audit code of a record.

By identifying individual sentences in the admission record's note the system aims to predict the best audit code per sentence, with the expectation that there is likely to be at least one sentence that will deliver the currently assigned audit code, after which any other audit codes can be assessed for usefulness and likelihood for reporting of additional codes. To illustrate this and the preceding point about individual sentences not necessarily matching the audit code of the note, consider the examples in

Table 4-5:

Table 4-5 Deriving Audit Codes per Sentence

No.	Assigned Audit Code	Notes	Sentences	Rule-Based Prediction	SVM Prediction
1	CRANIAL:TRAUMA:SAH	Transfer from XXXX. Small L tempoparietal EDH, R frontal and temporal contusions, R parietal#, SAH	Transfer from XXXX Small L tempoparietal EDH R frontal temporal contusions R parietal # SAH	(None - eliminated as words are unidentified) CRANIAL:TRAUMA:EDH (None - but retained as words are diagnostic) CRANIAL:TRAUMA:CONTUSIONS CRANIAL:TRAUMA:SKULL FRACTURE CRANIAL:TRAUMA:SAH	CRANIAL:TRAUMA:SDH CRANIAL:TRAUMA:SKULL FRACTURE CRANIAL:TRAUMA:CONTUSIONS SPINE:TRAUMA:FRACTURE CRANIAL:TRAUMA:SAH
2	CRANIAL:TRAUMA:EDH	R frontoparietal EDH. Intoxicated fall w HS	R frontoparietal EDH Intoxicated fall w HS	CRANIAL:TRAUMA:EDH (None - but retained as words are traumatic)	CRANIAL:TRAUMA:EDH CRANIAL:TRAUMA:TBI
3	CRANIAL:TRAUMA:CONTUSIONS	Inferior left frontal lobe cerebral contusions and haematoma	Inferior left frontal lobe cerebral contusions haematoma	CRANIAL:TRAUMA:CONTUSIONS CRANIAL:TRAUMA	CRANIAL:TRAUMA COMPLICATION:POSTOP BLEED
4	CRANIAL:TRAUMA:SKULL FRACTURE	Bilateral PTB/clinoid #. Surfing accident. No carotid injury. No intracranil haem.	Bilateral PTB/clinoid # Surfing accident No intracranial haem	CRANIAL:TRAUMA:SKULL FRACTURE (None - but retained as words are traumatic) *Negated* CRANIAL:TRAUMA:ICH	CRANIAL:TRAUMA:SKULL FRACTURE CRANIAL:TRAUMA CRANIAL:TRAUMA:SAH
5	CRANIAL:ANEURYSM (UNRUPTURED)	Bilateral MCA aneurysms. ?vasculitis	Bilateral MCA aneurysms ? vasculitis	CRANIAL:ANEURYSM probable COMPLICATION:INFECTION	CRANIAL:ANEURYSM (UNRUPTURED) CRANIAL:OTHER

The assigned audit code is the single audit code that has arrived with the incoming note, which is the complete text of the record. Sentences are derived from notes based on the presence of a comma or full-stop, or the word *and*. As described earlier certain stop words are removed from the sentences, and ambiguities and critical spelling errors are resolved. Each sentence is processed by the rule-based system to identify key words, some of which will be directly useful for labelling audit codes, others will be found to be concept identification words (such as those used for describing parts of the body, tests, negations, or a traumatic event, etc.), and some sentences will contain no useful words. After the rule-based system has completed its work the sentences identified as useful, together with the incoming assigned audit code, are given to the ML predictive systems. In the first instance the ML systems independently use the data as training and test sets, but after a ML system is incorporated into the main application it will return its predictions as part of the workflow of the application.

By splitting the note into individual sentences, and in the case of the rule-based system by looking for specific audit code words even within each sentence, the end-result is that multiple sentence-

specific classifications are derived. However, when evaluating these the only available comparative class is the incoming audit code, so if sentences and words are discovered which return the incoming audit code then they are ranked to return the primary result, and all other sentences and words may then be used to return any additional codes.

In the first example in

Table 4-5, no useful words were identified in the first sentence (transfer from another, de-identified, hospital), so it is not passed to the ML process. All the other sentences are used to train (or test) the ML process as indicative of the incoming audit code CRANIAL:TRAUMA:SAH. Although as components of the entire record they can be considered as indicative of SAH, on a sentence-by-sentence level they more clearly point to other audit codes. This is evidenced by the choices made by both the rule-based and SVM-based systems, which only agree once with SAH as a code. The accompanying Note examples 2 through 4 illustrate how the individual sentences from the first example contain words similar to those that are more typically associated with audit codes other than SAH, the rule-based and the machine learned SVM-based systems are identifying these codes.

In the example only one sentence is clearly indicative of SAH, which is the very last single-word sentence of the Note: SAH. In the system's design there is no requirement for reporting the already assigned SAH, so that sentence can be used to confirm an accurate prediction for the record, but apart from that has no purpose. The other sentences that do not agree with the incoming code are candidates for reporting as additional or more correct codes, even though when compared against the incoming SAH code they are inaccurate. The ML based system can also be fed the entire note for a single prediction, when supplied this Note the SVM predicted CRANIAL:TRAUMA:CONTUSIONS.

To conclude, when evaluating the prediction performance of the systems, multiple audit codes are derived, and since these can only be compared against a single incoming audit code many of them have the effect of reducing the reported accuracy of the system. It is expected however that

these will provide the core data required for the system's task of alerting the audit coder of possible additional codes.

4.9 Reporting Additional Audit Codes

After identifying any sentences that match the already assigned audit code the design of the system is to report all relevant *additional* codes, a mechanism which depends upon a further processing step of filtering and ranking the codes. In many cases these additional codes may be less precise, or constitute an alternative and therefore equivalent code to the main one, so they should not be reported. In other cases, they may constitute a completely unlikely code and should also not be used, but aside from these exceptions they should be ranked as most relevant to the main audit code and reported on.

In the example in

Table 4-5 EDH and CONTUSIONS should most likely be reported, and perhaps the SKULL FRACTURE, but it may well be that some of these may be considered irrelevant compared to the more severe SAH of the main record – however this is a decision that should be deferred to an expert being notified of the possible additional codes, or to an expert system still to be developed. The SVM-based suggestion of SPINE:TRAUMA:FRACTURE for the 5th sentence is based only on the statistical likelihood of words in the containing sentence, in fact "parietal #" is indicative of skull fracture, which the rule-based system has delivered, not of spinal fracture. Likewise, the 3rd sentence that reads "R frontal" is predicted by the SVM as CRANIAL:TRAUMA:SKULL FRACTURE based on statistical likelihood, but the rule-based system has decided that it has insufficient information to predict anything here. This is also the case with the sentences "Intoxicated fall w HS" and "Surfing accident" of the 2nd and 4th examples – the SVM system has predicted CRANIAL:TRAUMA:TBI and CRANIAL:TRAUMA – these may be statistically likely but the rule-based system has no rule for deriving these audit codes, as the nature of the information in them means that they cannot justifiably be the basis of a qualified prediction.

The 2nd and 3rd records contain examples of possible predicted codes that should *not* be reported, as they are either less precise but similar to the incoming audit code (CRANIAL:TRAUMA is less precise but of the same family as CRANIAL:TRAUMA:CONTUSIONS) or unlikely to be correct (COMPLICATION:POSTOP BLEED in the context of CRANIAL:TRAUMA:CONTUSIONS). Others such as the SVM system's predicted CRANIAL:TRAUMA:TBI in the 2nd record may be reportable, but as this is derived based on statistical likelihood rather than rule-based word matching it is perhaps inaccurate. In any case a process of eliminating similar codes may decide that the TBI need not be delivered in addition to the already existing EDH. Finally, the 4th record contains the phrase "No intracranial haem" that might be interpreted as an intracranial haemorrhage (the ICH found by the rule-based system) or sub-arachnoid haemorrhage (the SAH found by the SVM), but because of the negation the rule-based system would not report it.

The 5th record is an example of where the SVM-based system has delivered a more accurate audit code than the rule-based system: The sentence "Bilateral MCA aneurysms" has been correctly identified as CRANIAL:ANEURYSM (UNRUPTURED), whereas the rule-based system was only certain of CRANIAL:ANEURYSM. Upon analysis of this and other un-ruptured aneurysm codes delivered by the SVM-based system a pattern would no doubt emerge about how to predict an aneurysm as being un-ruptured, and this could be incorporated into the rules for this code, but at the moment only the SVM-system has been able to detect this and deliver a correct value. However, despite this accuracy, in reality either of these codes would be an adequate match to the incoming code, leaving only the additional sentence "? vasculitis" as a potential additional code – in this case the rule-based system has more accurately identified this as *probable* COMPLICATION:INFECTION whereas the SVM-based system could only achieve CRANIAL:OTHER.

In summary there must be a subsequent rule-based process to refine the list of additional codes before they are reported on, a step that will increase the accuracy of the system by a process of elimination and ranking. Codes can be taken from either the rule-based system or the ML SVM-based one, according to their likely accuracy, which will also need to be assessed algorithmically. Care should be exercised when using ML-based statistically derived codes, if there is no instance of a codifiable word in a sentence it is unlikely to be useful to predict an audit code for it.

4.10 Combining Prediction systems

As described in the previous section, the system must be able to derive the best outcome by combining rule-based with machine learning-based predictions, and the mechanism for achieving this is itself rule-based; but in order to scope this the accuracy of the different systems must be compared, to enable decisions about which codes are more amenable to combining predictions. To accomplish this, predictions were made for all classes over all of the data, which are tabulated with the precision, recall and F-measure of each of the rule-based and SVM-based classifications compared – based on sentence matching against incoming note-based audit codes, as discussed earlier.

In order to fairly compare the two systems any admission records that were assigned an incoming audit code of OTHER, or were unclassified (that is, having the diagnosis "Cranial" which has not been assigned an audit code), were removed from the data, as were any predictions of OTHER. The system is being trained to predict valid audit codes where possible, and only uses OTHER as a "last resort" code when alternative predictions are not found, so it's not useful to train it to suggest OTHER, or to compare accuracy in predicting it. Some SVM-based predictions of OTHER were retained, where eliminating the SVM prediction would have involved removing a legitimate audit code coming from the linked rule-based prediction. The total number of records used in the comparative evaluation was 12,023. The table is reproduced in Appendix A, and the summary Table 4-7 in Section 4.11 accompanies a discussion of the results. A separate comparison was performed on the predictions obtained by each system for the OTHER and unclassified records, and those that the rule-based system had failed to predict. Tables of these results are reproduced in Appendix B, with a discussion in Section 4.12.

The measurements are performed using both micro and macro-averaging, with the latter favoured for its giving weight to classes equally. Additionally, a combined score is calculated by a combination of the predictions – using the rule-based then adding (through programmatic use of an OR clause) the SVM-based predictions and their matches. A decision can then be made about

the best approach – if combining the systems produces a better F-score for a code then it is useful, otherwise only the rule-based system is used, as it is explainable and able to be programmatically tuned. The final system is a hybrid of both combined and rule-based predictions, and is therefore labelled as a Hybrid system.

Combining the systems results an increase in matches, but also in more predictions than the actual instances, so the precision certainly decreases. A limited number of codes show an overall improvement by combining the two prediction methods, these are shown as "Combined" in the Hybrid F-Score Source column of the table. To illustrate, the effects of combining the predictions from both systems is explored for the COMPLICATIONS and CRANIAL:TRAUMA:SKULL FRACTURE codes:

Table 4-6 Combined Prediction Systems

DiagnosisAuditCode	Class Total	Rule Based Matched	Rule Based Predicted	Rule Based FScore	SVM Based Matched	SVM Based Predicted	SVM Based FScore	Combined Matched	Combined Predicted	Combined FScore	Combined - Rule based Difference
COMPLICATION	465	144	171	0.453	170	224	0.494	223	300	0.583	0.130
CRANIAL:TRAUMA:SK	369	322	443	0.793	327	751	0.584	352	838	0.583	-0.210

F-score ranks the ratio between what is predicted and what is actually correct, which for COMPLICATIONS is a little lower in the rule-based system compared with the SVM-based system, but when the systems are combined the F-score increases by 13%. By contrast the rule-based F-score for CRANIAL:TRAUMA:SKULL FRACTURE is considerably better than the SVM-based score, and the combined F-score, while coincidentally equal to that achieved by the combining for COMPLICATIONS, is in fact 21% less that the F-score achieved by the rule-based system alone for CRANIAL:TRAUMA:SKULL FRACTURE. The reason for these differences are because of changes in the ratio between true and false predictions – as the sets are combined there is some gain in the number of matches but accompanying this is the combined gain in predictions. For a combination to be worthwhile the increase in false predictions must not outweigh the increase in true predictions (matches) achieved.

These numbers can be visualised using Venn diagrams – for instance to illustrate the ratios between predictions and matches for the SVM-based system for COMPLICATIONS, there are a total of 224 predictions with 170 of them being true predictions (matches), meaning 54 false predictions (224 - 170 = 54), shown as overlapping areas in Figure 4.1.

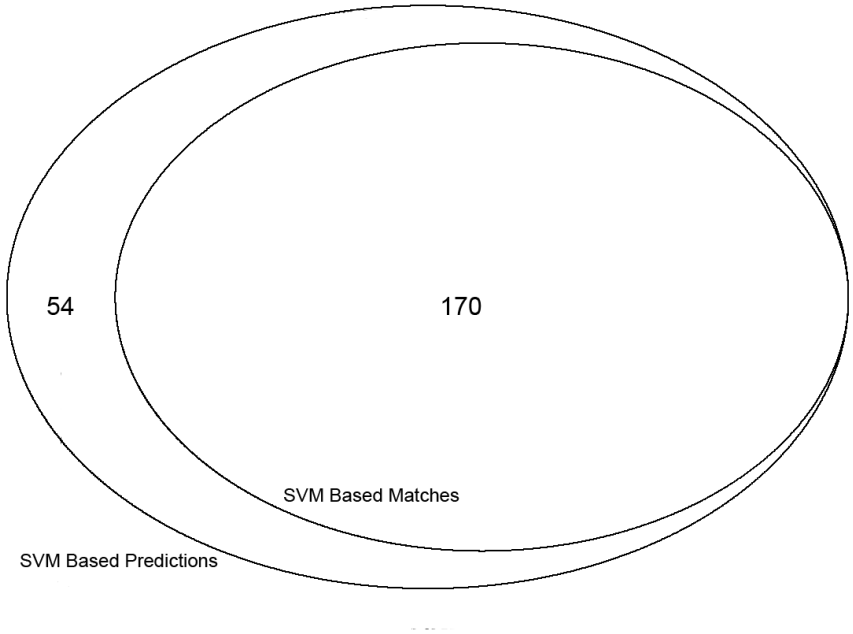


Figure 4.1 COMPLICATIONS audit code: SVM-based predictions vs matches

For the rule-based system there are 171 predictions with 144 being true predictions, leaving 27 false predictions - show in Figure 4.2.

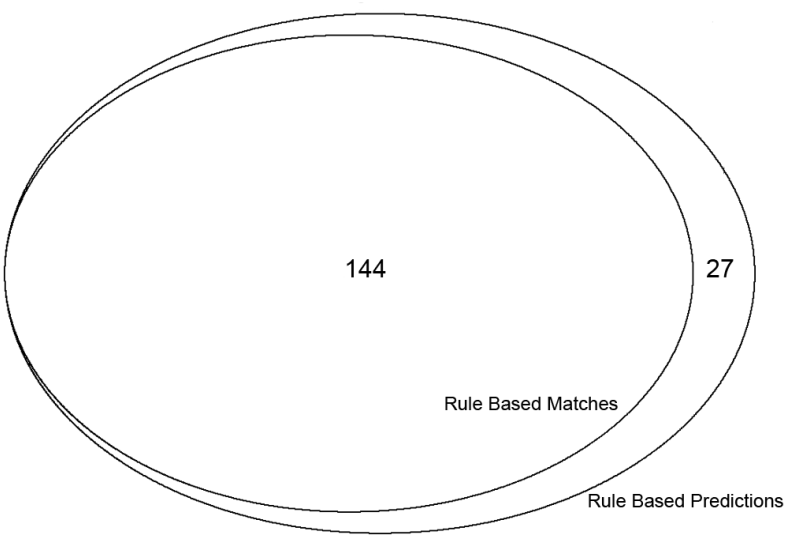


Figure 4.2 COMPLICATIONS audit code: Rule-based predictions vs matches

Combining these sets into a new Venn diagram (Figure 4.3) shows that there are 91 true predictions (matches) in common, with the remaining predictions from each system combining to increase the totals for both true predictions (total matches) and false predictions. The Venn diagram shows the combined scores for each component: the SVM-based matches are split as the matches total of 170 minus the 91 in common, resulting in components of 79 and 91; the rule-based matches are split as the original 144 minus 91 = 53, and 91. The combined total of matches from both systems is 79 + 91 + 53 = 223. Likewise, the remaining combined false predictions are displayed as excess numbers 50, 23, 2 and 2 based on the combinations and overlaps, meaning 77 false predictions over the total of 300 predictions. Comparing rule-based to combined numbers, the increase of false predictions from 27 to 77 (50) is outweighed by the increase in true predictions from 144 to 223 (79), so the combination is worthwhile.

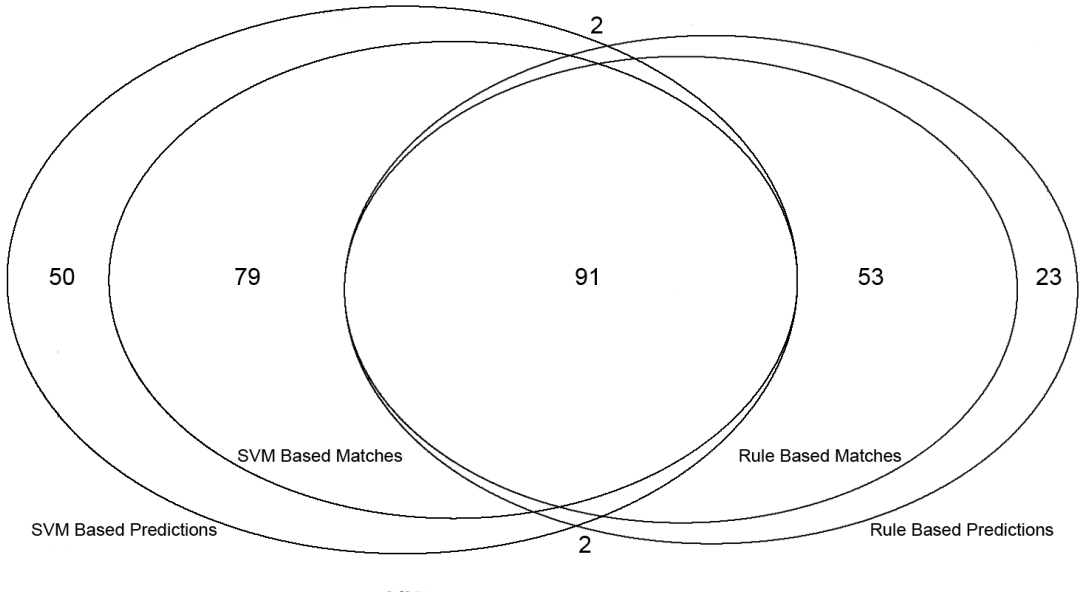


Figure 4.3 COMPLICATIONS audit code: Combined SVM and rule-based predictions

By contrast, in Figure 4.4 the overlapping sets for CRANIAL:TRAUMA:SKULL FRACTURE show that the increase in false predictions of 365 resulting from the combination vastly outweighs the 30 increase for true predictions, so the combination is not worthwhile:

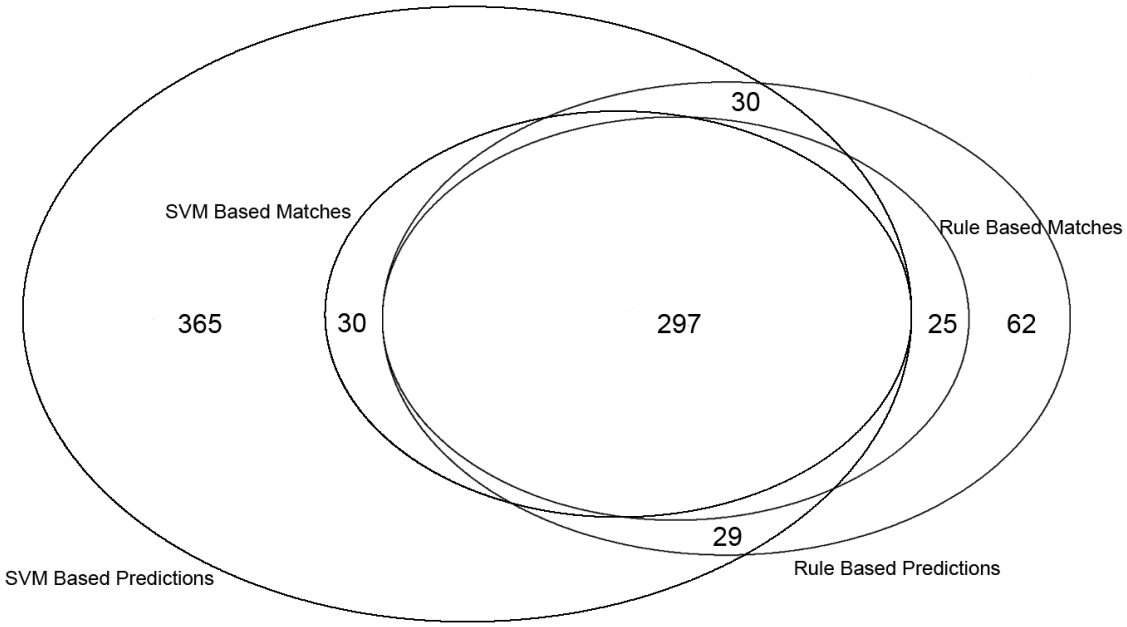


Figure 4.4 CRANIAL:TRAUMA:SKULL FRACTURE audit code: Combined predictions

Deciding on which predicted audit codes can be reported as additional codes is a programmatic step, and part of the process is to filter out obviously unnecessary codes – for reasons of either their duplication of the incoming code or their inappropriateness compared to the incoming code. In other words, the system will remove inaccuracies as a further step, and so the inaccuracies of any combined systems will also be reduced; however, a combination like the CRANIAL:TRAUMA:SKULL FRACTURE discussed here would likely not be worth the effort to implement.

4.11 Evaluation Results

The complete figures are available in the Appendices A and B, but a summary of them is in Table 4-7 below. The four groupings are the results from Rule-Based predictions; the results of Sequential Minimal Optimization SVM Based predictions; the results from combining them with

a logical OR; and the final results obtained by taking only the good combinations (those resulting in an increase in F-measure) together with the default rule-based predictions.

The Hybrid group of the table shows scores after taking only the combinations that contribute positively to an increase in F-measure, with the remainder being supplied by the rule-based approach, there is an increase in recall from 75.4% to 78.0%. However, there is an accompanying decrease in precision from 77.7% to 74.6% offsetting this gain, resulting in an overall increase in F-measure of only 0.3% when calculating by using macro weighted averages, and 1.4% when using micro averages.

Table 4-7 Summary of results

		Rule Based			SVM Based			Combined			Hybrid		
	Class	Matched	Predicted		Matched	Predicted		Matched	Predicted		Matched	Predicted	
Totals	12023	8159	12023		7631	12023		9471	17176		8913	13685	
		Precision	Recall	F Score	Precision	Recall	F Score	Precision	Recall	F Score	Precision	Recall	F Score
Micro averages		0.679	0.679	0.679	0.635	0.635	0.635	0.551	0.788	0.649	0.651	0.741	0.693
Macro Weighted averages		0.777	0.754	0.749	0.695	0.680	0.672	0.643	0.827	0.709	0.746	0.780	0.751

Note that the total audit code instances being processed is 12,023 – which is the same as the total predictions for each of the rule-based and SVM-based systems on their own – every record is predicted once under each scenario. Consequently, the micro averages for Precision, Recall, and F-measure are identical within each system (since the micro average calculation for Precision is Total Matches/Total Predictions, while the micro average calculation for Recall is Total Matches/Class Total, but Total Predictions and Class Total are identical). However, when combining the data by allowing a logical rule-based *OR* SVM-based match the result is more predictions than there are classes (a prediction can be for *this* class OR for *that* class), with an accompanying decrease in precision offsetting the increased recall (matches) obtained. Also note that macro averaging takes the individually calculated scores for precision, recall and F-measure, extended by the matches in each class, sums these, and then divides those totals by the total of the matches - which means that these figures represent the accuracy of each class proportionally. There is a more detailed explanation of the different approaches in Appendix D.

4.12 OTHER and Unclassified Records

The rule-based component never tries to predict the audit code OTHER, and only uses OTHER when no other audit code can be identified. Accordingly, the SVM system is not learned to predict OTHER, though in a special case outlined in the next section, it also may return OTHER as a default. For this reason, rule-based predictions of OTHER and their accompanying SVM-based predictions were excluded from the main comparative evaluation of the two approaches. Nonetheless, when the rule-based component cannot predict anything (and so predicts OTHER), then the only alternative code is the SVM-based one, and in this scenario it performs rather well, matching 69.7% of the records, with a precision of 90.6% - see Appendix B. There is an overall decrease in all scores when these records are included in the main comparison because of the extra imprecision of the rule-based component (incorrectly predicting OTHER) and the extra overall unmatched SVM-based predictions, this can be seen in the additional tables in Appendix B. These results indicate that the SVM-based prediction should *replace* that of the rule-based one when there is no valid suggestion from the rule-based component – using a logical 'IF THEN' instead of an OR: "*If* rule-based prediction is OTHER *then* SVM-based prediction", rather than the usual "rule-based *or* SVM-based prediction" approach.

For the same reason, that the rule-based system never tries to predict OTHER, when the actual incoming audit code is OTHER there is no point in assessing the accuracy of predicting it. In the situation where OTHER is the only valid audit code because of a genuine lack of a suitable audit code for the incoming record, and no alternative audit code is discovered by the prediction system, then predicting the default of OTHER only confirms the incoming record. And confirming OTHER contributes nothing towards the system's function of alerting the auditor about unclassified data.

The situation is repeated when the incoming record's diagnosis is a simple "Cranial" – this means that the diagnosis is not linked to a valid audit code, and there is no point in confirming this lack of audit code by trying to predict it – the system should predict a valid alternative if possible. In this case the system is assigning the incoming record a default audit code of CRANIAL:UNCLASSIFIED in order to label it, but it should never try to predict this label. For the

rule-based system, as with the OTHER incoming code, using this label would only be as a last resort to confirm that no alternative has been found. The SVM-based system would no doubt try to learn for it – but in any case, a prediction of CRANIAL:UNCLASSIFIED is not at all useful. Instead, incoming audit codes of OTHER or CRANIAL:UNCLASSIFIED should be more usefully evaluated by the system's distribution of them into *alternative* audit code predictions - they should ideally be classified entirely as additional or alternative audit codes.

Because of the objective to re-classify the audit code these predicted codes cannot be assessed for accuracy unless by a system expert evaluating their correctness, which is also true of the predicted additional audit codes of any other record – they differ from the incoming code and so cannot be verified internally against the incoming code. Nevertheless, based on the accuracy of predictions for these audit codes measured when they *are* the incoming audit code, their accuracy may be inferred. Appendix C summarises these records, showing counts and percentages of matches into various audit codes by the rule-based and SVM-based methods for incoming OTHER and CRANIAL:UNCLASSIFIED records. The tables also include a column recording the counts and percentages where both systems are in agreement on the prediction. Using the rule-based method almost 78% of incoming OTHER audit code records are re-classified, while the SVM-based method achieves around 70%, and the two methods agree with one another 36% of the time. For CRANIAL:UNCLASSIFIED the rule-based method is able to re-classify 91% of the records, the SVM-based method manages 77%, and they have 42% of their predictions in common.

4.13 Expert Evaluation

A domain expert was able to evaluate 100 predictions and make comments. Ten of these are produced below, together with comments by the researcher. Just over half of the codes were considered correct, although some of these raised questions by the expert about the audit coding structure in the department. The rest of the codes were considered incorrect by the domain expert, but many of these did in fact consistently predict the incoming audit code. The reason for rejecting the predicted codes was often related to the expert drawing a conclusion about what the final audit code should be, which is a conclusion an auditor would reach upon examination of all of the data.

The system is designed to bring to the auditor's attention all of the codifiable words, so that informed conclusions may be reached by the auditor.

Table 4-8 Example of Domain Experts evaluation

#	Notes	Diagnosis Audit Codes	Suggested Codes
	Domain Expert's Comments		Researcher's Comments
1	R occipital infarct, p/w GTCS, a/w MR	CRANIAL:VASCULAR	CRANIAL:VASCULAR:OCCLUSION
	FYI: GTCS = "Generalised tonic-clonic seizure"		GTCS is in the dictionary
2	Ped vs car. GCS 13. Small left EDH and SDH. Right traumatic SAH.	CRANIAL:TRAUMA:EDH	CRANIAL:TRAUMA:EDH CRANIAL:TRAUMA:SAH CRANIAL:TRAUMA:SDH
	Perfect!		
3	Grade I SAH	CRANIAL:SAH (NON-ANEURYSMAL)	CRANIAL:VASCULAR:SAH
	Correct		VASCULAR:SAH was an acceptable match to SAH (NON-ANEURYSMAL).
4	post-op transient dysphasia, improving	COMPLICATION	COMPLICATION COMPLICATION:NEUROLOGICAL
	Check whether a term like "improving" should be used as a flag for either following up to confirm that it persisted (or didn't), or whether it can simply be used to exclude on the basis that it probably wasn't permanent		Cannot derive a generalizable rule from a mention of "improving". The auditor needs to make a choice whether or not to use the code.
5	mild TBI, traumatic SAH, seizure, L) SDH	CRANIAL:TRAUMA:TBI	CRANIAL:TRAUMA:SAH CRANIAL:TRAUMA:SDH CRANIAL:TRAUMA:TBI EPILEPSY
	A post-traumatic seizure should probably not trigger a code for "epilepsy", as the patient probably wasn't epileptic before or after the TBI. Also, is TBI not categorised as mild/moderate/severe in the annual audit? If it is, then this needs to be preserved in the coding.		Need a rule to eliminate linking of seizure with epilepsy in context of a CRANIAL:TRAUMA. Mild, moderate, and severe all get coded as CRANIAL:TRAUMA:TBI in the coding system provided. The annual audit does distinguish these, so the coding system should be altered.

Table 4-8 Example of Domain Experts evaluation

#	Notes	Diagnosis Audit Codes	Suggested Codes
	Domain Expert's Comments		Researcher's Comments
6	?VP shunt malfunction	CRANIAL:OTHER	CRANIAL:SHUNT
	Blocked / malfunctioning shunts should be coded separately to simple "shunts"		There is no simple SHUNT, only CRANIAL:SHUNT.
7	bilateral leg weakness, CT no #, For MRI	PAIN	COMPLICATION:NEUROLOGICAL
	Difficult to be certain - main diagnosis is coded as cranial but notes suggest spinal (leg weakness + pain suggests spinal nerve root compression, so a radiculopathy)		There is a rule in place for the combination of leg weakness and pain to be interpreted as radiculopathy (SPINE: DEGENERATIVE), but at this point of our evaluation we are predicting only from Notes, and in this instance there is no pain information in the Notes, PAIN is coming from the incoming audit code only.
8	L facial nerve palsy post-op	COMPLICATION	COMPLICATION COMPLICATION:NEUROLOGICAL PERIPHERAL NERVE
	Facial nerve is a cranial nerve, so it's not strictly a peripheral nerve complication		The incoming audit code was correctly predicted. Regarding the PERIPHERAL NERVE suggestion, there is no FACIAL NERVE audit code. Facial nerve issues variously get the diagnosis Cranial>Pain>Facial> Trigeminal Neuralgia, with associated audit code of PAIN; or as Cranial>Other (CRANIAL:OTHER); or as COMPLICATION. Is there a need for another code or should we just eliminate other possible suggestions if one of these is used to categorize a facial nerve condition?
9	MBA GCS 12. small amount of intraventricular blood	CRANIAL:TRAUMA:IVH	CRANIAL:TRAUMA:IVH
	TBI		The incoming audit code was correctly predicted. Is there a rule that can be applied for a mention of intraventricular blood (IVH) to be instead coded as TBI? It seems that only the auditor can make a

Table 4-8 Example of Domain Experts evaluation

#	Notes	Diagnosis Audit Codes	Suggested Codes
	Domain Expert's Comments		Researcher's Comments
			decision to code as TBI when IVH is clearly indicated in the Notes.
10	GCS 10, post fall. L frontotemporal ICH, SDH, SAH	CRANIAL:TRAUMA:ICH	CRANIAL:TRAUMA:ICH CRANIAL:TRAUMA:SAH CRANIAL:TRAUMA:SDH
	TBI		Likewise here - the incoming audit code was correctly predicted: What is needed here is to know whether it is in fact correct to draw out all 3 codes, that is - to suggest the SDH and SAH as additional to the category of ICH already given. Instead the suggested code is TBI. Is there a rule that can be applied to substitute TBI for a matrix of ICH, SDH, and SAH?

Examples 1 to 3 are where the domain expert is satisfied with the predicted audit code. Note that in example 3 the predicted code differs from the incoming code but is close enough (both are non-traumatic SAH). Example 4 is a case where the expert has suggested that a rule could be derived based on the appearance of the word "improving", but the researcher observes that this is not generalizable; this is because there is no consistency in its use, and words like this are more likely to be useful to the auditor on a case by case basis. Examples 5 and 6 are instances where a revision of the existing audit coding structure is indicated. Example 7 shows that the expert expects the system to take account of the incoming audit code (PAIN) when predicting additional codes, and the system must do that – but at this stage of the evaluation all predictions were only proceeding from the Notes, as described by the researcher's comment. Example 8 is a situation where the existing data is confusing in its variety so a rule definitely would provide clarity.

As an example of where the domain expert has indicated a different final audit code rather than confirm the correctness of the incoming or predicted audit codes, compare the results of example 2 with those of examples 9 and 10. In example 2 the system finds three audit codes - EDH, SAH, and SDH. These are all types of brain haemorrhage, and one of them, EDH, agrees with the incoming audit code. The domain expert has said that this is a perfect result. However, in example 9 the system just as accurately discovers IVH, another type of brain haemorrhage and in agreement with the incoming audit code, but the expert has said that it should be interpreted as TBI – traumatic brain injury. Likewise, in example 10 the system finds the clearly discoverable trio of brain haemorrhages of ICH, SDH, and SAH, with ICH agreeing with the incoming audit code; but the expert seems to indicate that this should have been interpreted as TBI, or perhaps as TBI in addition to the identified codes. It seems possible that TBI is an overarching code that in some cases should be used to summarize a number of other codes, and the system does not yet have a rule that handles the scenarios in which this must occur. Nevertheless, it seems safer to leave this sort of decision with the auditor – just bringing these to his or her attention fulfils the purpose of the system.

Example 7 has a comment from the domain expert that "leg weakness + pain suggests spinal nerve root compression, so a radiculopathy". This type of deduction is typically the approach taken by the domain expert, and is the reason why a rule-based system was concluded to be the best fit for the neurosurgical department. The expert expects to be able to make deductions based on a set of conditions, and these should be patterns that are likewise embedded in the predictions that the system makes. In this case the rule actually was in place but the required pain component of the rule was missing from the Note, it could have only been derived from the incoming audit code, which at this stage of the evaluation process was not included. The domain expert's expectation clearly shows that the eventual predictive system must also take account of the incoming audit code and diagnosis, and not use only the Notes as its data source.

For points 9 and 10 the expert has not provided any detail of the conditions used to deduce TBI, so at this point no rules can be introduced to the system for this.

4.14 Increasing the Accuracy

By combining rule-based and machine-learning based methods (see Table 4-7) it is possible to increase the recall of the system from the rule-based method's 75.4% to the combined method's result of 82.7%, but this comes with a large decrease in precision, dropping from 77.7% for the rule-based approach to 65.3%. The differences in F-measure reflect the overall result – it drops four percent, from 74.9% to 70.9%. By using only those combinations which yield an F-measure increase, the system can increase recall to 78.0% (from 75.4%) and only drop precision down to 74.5% (from 77.7%), resulting in a slight increase of F-measure to 75.1% from the original 74.9%. Increasing the precision or the recall will benefit the overall score; recall can be increased by adding more rules or further learning of the ML method, but precision can be increased by eliminating obviously incorrect predictions.

The SVM-based ML method can be corrected by checking when certain predictions are suggested and modifying the input before re-testing the prediction – a good example of where this works is the prediction for audit code CARPAL TUNNEL. Because of the limited number of words used in the notes of carpal tunnel admissions – phrases like "left CTS", "right sided", "R) carpal tunnel", "L hand", "left", "right", "Right carpal tunnel entrapment" etc., there is a preponderance of words that indicate left and right. In a standard NLP system these words might be considered as "stop words" and eliminated from consideration but here they need to be retained as they are important statistically as well as for the information they convey. Unfortunately, the effect of the importance of these words for carpal tunnel means that other records can get classified as carpal tunnel when they also contain words indicating left and right. There is a total of 81 records in the current data (plus two classified as OTHER) where the incoming audit code is carpal tunnel, but an unmodified SVM-based system predicts 859 matches, 635 of which are the primary prediction of a record; 65 of which are correct matches – yielding a mere 10.2% precision. By contrast an unmodified rule-based system requires other important words like "hand", "CTS", "CTR", and "carpal", and makes only 54 predictions; 53 of which are the primary prediction, 51 of which are matches – giving 94.5% precision and 62.9% recall.

Because the SVM-based method is integrated into the system it can be programmatically modified, and if carpal tunnel is predicted then it can be checked by looking to see if the incoming text contains any carpal tunnel significant terms like “CTS”, “carpal”, “tunnel”, and if not but it does contain a left or right type of word together with some other words, then the left and right words can be stripped out of the text and the phrase re-submitted. As a result, the incorrect weighting given to the left or right is put aside and another prediction can emerge. By adopting this approach, the predictions decrease from 859 to 203, with 163 of them being the primary prediction; and 65 correct matches over 163 predictions improves precision to 39.9%. If after conducting this exercise the SVM no longer predicts anything, then as with the rule-based system when a suitable code cannot be identified, a default code of OTHER is returned, the only time the SVM system does this.

The rule-based system is easily amenable to further adjustments, and in the case of carpal tunnel even though the rule requires an identifiable term like “carpal”, “CTR” etc. and it won’t predict carpal tunnel just for a single word like “left”, it can be programmed to just use the incoming audit code if it has found no classifiable text at the end of its processing, and the incoming text is just one sentence containing one of these identifiers. As the aim of the system is to find additional codes, if no audit codes are found at all then there is no harm in just confirming the incoming code if the note seems to contain enough information – with the effect of eliminating any uncertainty around these records and confirming that no more additional codes need to be suggested. By adopting this approach, a further 31 cases are added to the rule-based system, its predicted count increases to 85, with 81 matches – 95.3% precision with 100% recall.

Regarding those rule-based carpal tunnel predictions that don’t match the incoming audit code, with the addition of a further 2 found when including the OTHER audit code, the incoming audit codes and accompanying note combination suggest that in many cases CARPAL TUNNEL would have been the correct audit code – in the Table 4-9 it is likely that all except COMPLICATION:INFECTION could have been more correctly classified as CARPAL TUNNEL, and certainly they should be suggested as alternatives.

Table 4-9 Carpal Tunnel main predictions where incoming code differs

Incoming Audit Code	Notes
COMPLICATION:INFECTION	Carpal tunnel
OTHER	Left Carpal tunnel release
OTHER	Left carpal tunnel syndrome, left cubital tunnel syndrome
PERIPHERAL NERVE	L carpal tunnel
PERIPHERAL NERVE	Right carpal tunnel decompression
PERIPHERAL NERVE	Right carpal tunnel syndrome

Further precision can also be obtained for both methods by performing comparisons between the incoming audit code and the codes predicted by the method; and also by checking whether the text being predicted upon even holds sufficient information for a proper prediction. Sometimes a comparison will conclude that a predicted code is unlikely to be found in conjunction with the incoming code or the other predicted codes, sometimes the text is simply inadequate to predict meaningfully, and at other times a prediction may be likely but it will be redundant to report it. In these cases, retaining the prediction would only decrease precision with no correlated increase in recall or useful additional codes, and so the predicted code can be suppressed.

The Table 4-10 illustrates some of these, starting again with CARPAL TUNNEL, with the first entry being an example where CARPAL TUNNEL is an accurate additional prediction that should be retained. Some of the examples are SVM-based predictions where there was insufficient information to have another code emerge after removing the left or right words and re-predicting, either because there was no useful text or because there had not been sufficient learning on the text to make it recognizable. The column "Sentence predicted on" shows the actual text the system predicted on, sometimes this is just a single left or right word because of the way text is broken on commas and full-stops, and these are single word sentences as a result. In some of these the main text has been correctly classified; for instance, "Guyon's canal" was correctly classified as ULNAR NERVE. The column "Reason" will contain either "Valid" where the classification is clearly valid in the context of the incoming audit code, or "Possible" where it could valid - in these instances the code should be kept and so the linked Action will be to "Retain". The Reason column may contain other values where the resulting Action should be to "Discard": "Information" meaning insufficient information to predict from, "Invalid" meaning it cannot exist in the context

of the incoming audit code, "Redundant" meaning it is not required to report the code as there is already sufficient similar coverage in the incoming audit code.

The last four records show that a Note accompanying an incoming audit code of CRANIAL:TRAUMA:TBI was broken into four sentences and none of them are fit to suggest additional codes from, either because the incoming audit code is sufficient or the text used to predict from doesn't contain enough information to be able to rely on the prediction. In this case if the entire Notes is given to the SVM-based method it returns the same code as one of the sentences: CRANIAL:TRAUMA:CONTUSIONS, which is an accurate alternative to the incoming CRANIAL:TRAUMA:TBI code and serves to confirm the accuracy of the prediction. However, for reporting beyond this confirmation of accuracy the code should be discarded as being redundant, the incoming code has already covered this.

Table 4-10 Predicted audit codes filtering examples

Incoming Audit Code	Predicted Audit Code	Method	Primary / Additional	Reasons	Action	Notes	Sentence predicted on
SPINE:DEGENERATIVE	CARPAL TUNNEL	Rule	Additional	Valid	Retain	L3/4- L4/5 Laminectomy & Left CTR	Left CTR
ULNAR NERVE	CARPAL TUNNEL	SVM	Primary	Possible	Retain	Left cubital tunnel syndrome	Left cubital tunnel syndrome
ULNAR NERVE	CARPAL TUNNEL	SVM	Additional	Information	Discard	Right, Guyon's canal	right
CRANIAL:ANEURYSM	CARPAL TUNNEL	SVM	Primary	Information, Invalid	Discard	left	left
CRANIAL:NEOPLASIA:GLIOMA	CARPAL TUNNEL	SVM	Additional	Information, Invalid	Discard	Left, Grade IV GBM	Left
CRANIAL:TRAUMA:TBI	CRANIAL:TRAUMA	SVM	Primary	Redundant	Discard	CHI, GCS 3, ped v train, ETOH	CHI
CRANIAL:TRAUMA:TBI	CRANIAL:VASCULAR:SAH	SVM	Additional	Possible, Information	Discard	CHI, GCS 3, ped v train, ETOH	GCS 3
CRANIAL:TRAUMA:TBI	CRANIAL:TRAUMA:EDH	SVM	Additional	Redundant, Information	Discard	CHI, GCS 3, ped v train, ETOH	ped v train
CRANIAL:TRAUMA:TBI	CRANIAL:TRAUMA:CONTUSIONS	SVM	Additional	Information	Discard	CHI, GCS 3, ped v train, ETOH	ETOH

As the objective of this filtering process is to identify and remove redundant audit codes, including those that confirm the incoming code, there is no in-built mechanism for measuring the accuracy of the process, and it can only be confirmed as being accurate by the users of the system, by examining the proposed Audit Codes Report.

4.15 Challenges in Evaluation

A domain expert provided the initial classification rules in the form of existing SQL queries for the majority of the audit codes, then a number of the rules were subsequently developed during the research by observations of the data, in expectation of later verification by the expert.

However, feedback received by the domain expert in the latter stages of development covered only 100 examples, not enough to comprehensively verify the current rules. Consequently, the rules are not yet complete, in fact are always amenable to refinement. The system will need to be used and evaluated by the neurosurgical department in order to quantify its effectiveness in predicting additional and alternative codes.

When learning and evaluating the system, without a verified and complete set of rules or a gold-standard data set to train with, only the incoming audit code of records could be used as a standard, which presented a number of challenges:

The incoming audit code could only accurately apply to specific text in a Note, whereas there was often a lot of other classifiable text in a Note where a different audit code would be correct. Classifying this additional text is the main purpose of the system, but lacking an external standard meant that all of this text was included in the machine learning for the single incoming code. Despite this, there were usually sufficient examples where the text precisely matched the incoming code, so the ML process was able to deliver reasonably precise predictions.

In many instances there was ambiguity in matching of textual information to the audit classes - it was found that the same text was being classified against more than one audit code, for instance CRANIAL:TRAUMA:TBI and CRANIAL:TRAUMA:CONTUSIONS would both be used in the context of the key word *contusions* and therefore appear to be genuine alternative contenders for classification of contusions. To take account of this issue it was determined that either candidate prediction would suffice as a match, which freed the remaining text to be assessed for possible additional audit codes.

For some audit codes there were insufficient examples for machine learning to correctly learn against; for others the audit codes were often not even suitable to learn a system against – for instance diagnoses which mapped to the audit code OTHER, or to imprecise audit codes, or which had no mapped audit code. The diagnostic and audit coding systems allowed for different degrees of exactness according to the hierarchical structure of the codes, with the result that inexact

options were sometimes utilized when a better option could have been taken. Reclassifying these is one of the purposes of the system, with continuous use by the department these will be highlighted and will likely lead to refinement of the coding choices made.

For instance, the diagnostic code "Cranial" was often picked – it is the root of many more useful diagnostic codes, such as "Cranial>Trauma>Extraaxial>SDH" (which maps to audit code CRANIAL:TRAUMA:SDH), but "Cranial" by itself has not even been assigned an audit code and is therefore unclassifiable. Even the slightly more exact next member of the hierarchy "Cranial>Trauma" (which maps to audit code CRANIAL:TRAUMA), is not much more useful, and many records were assigned to these diagnoses when they should have been mapped to more useful codes.

It should be noted that there was sometimes a lack of a suitable audit code to precisely classify against, which meant that text that was capable of being classified exactly was instead relegated to a more general code, and consequently added imprecision to the system. For example, although there were many exact codes dealing with brain tumours, such as CRANIAL:NEOPLASIA:CAVERNOMA and CRANIAL:NEOPLASIA:MENINGIOMA - there were no detailed diagnosis and equivalent audit codes for corresponding spinal tumours (both cavernoma and meningioma are also found as spinal tumours) – all spinal neoplasia were classified together under SPINE:NEOPLASIA.

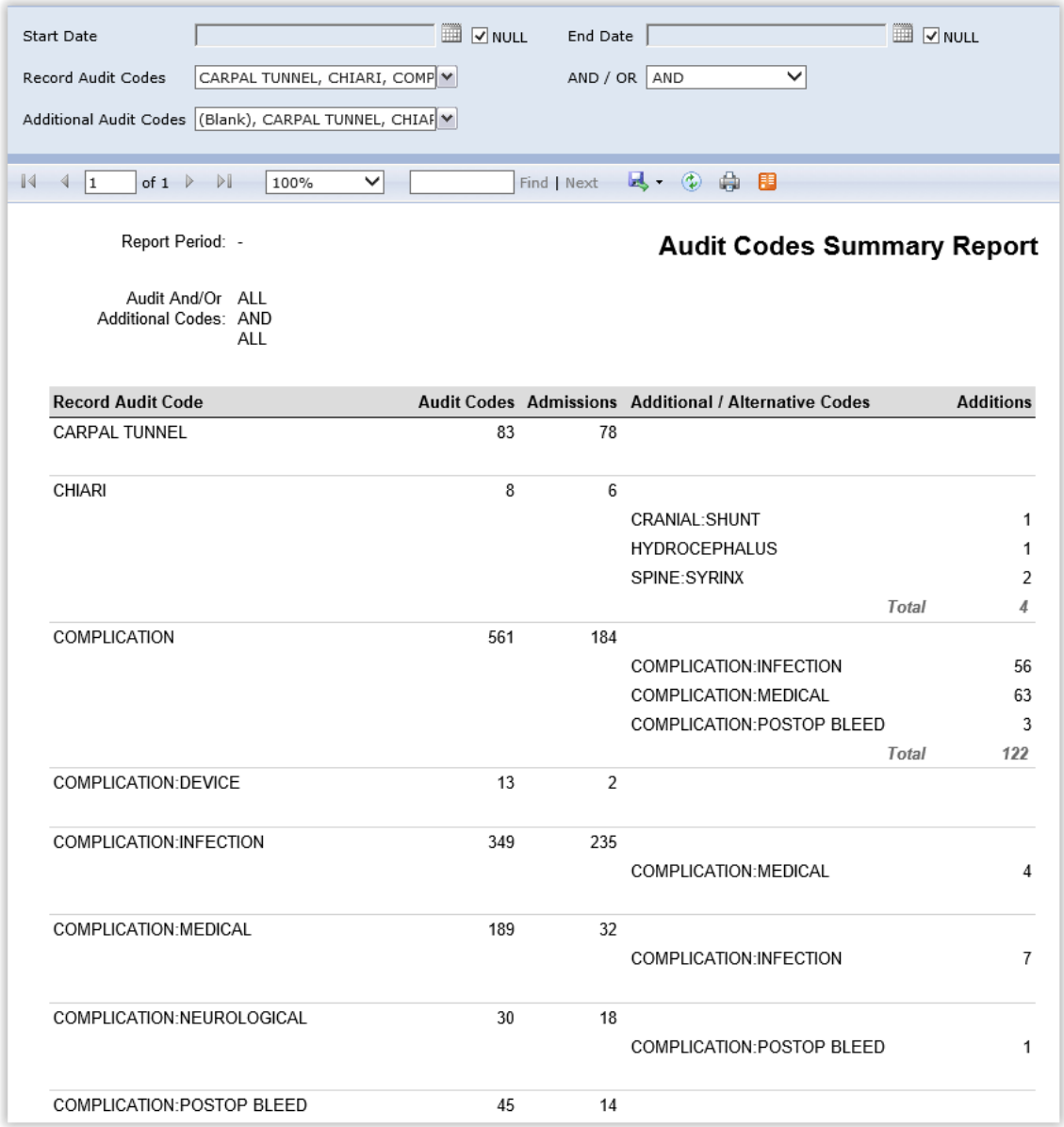
A limitation of the output of the system is that it can only return suggested audit codes deduced from the provided text, and although these can be cross-checked for suitability against other records for the same admission, there is no other intelligence built in. It is possible that certain combinations of codes would be better to be classified under an overarching code – for instance the presence of multiple injuries such as SDH, SAH, contusions etc. may possibly be better classified just as an overarching TBI (traumatic brain injury), as discussed above under Expert Evaluation. Therefore, the system reports on all the relevant codes, and anyone using the system will have to exercise judgement when reviewing the suggested additional codes – they are merely a reporting of additional codes discovered, not a recommendation that they are the correct final

audit codes. The Audit Codes Report will help an auditor to identify the relationships between the incoming audit code and suggested additional codes.

4.16 Proposed Audit Codes Report

This section describes the Audit Code Report presented as part of the architecture of the method, as shown in Figure 3.1 .The Audit Code Report proposed by the method is a browser-based SQL Server Reporting Services (SSRS) report, consisting of a Summary Report and a Detail Report, which allows an auditor to easily navigate the existing audit data and its predicted additional and alternative codes.

4.16.1 The Audit Codes Summary Report



Start Date NULL End Date NULL

Record Audit Codes CARPAL TUNNEL, CHIARI, COMP AND / OR AND

Additional Audit Codes (Blank), CARPAL TUNNEL, CHIAF

1 of 1 100% Find | Next

Report Period: -

Audit Codes Summary Report

Audit And/Or Additional Codes: ALL AND ALL

Record Audit Code	Audit Codes	Admissions	Additional / Alternative Codes	Additions
CARPAL TUNNEL	83	78		
CHIARI	8	6		
			CRANIAL:SHUNT	1
			HYDROCEPHALUS	1
			SPINE:SYRINX	2
			Total	*4*
COMPLICATION	561	184		
			COMPLICATION:INFECTION	56
			COMPLICATION:MEDICAL	63
			COMPLICATION:POSTOP BLEED	3
			Total	*122*
COMPLICATION:DEVICE	13	2		
COMPLICATION:INFECTION	349	235		
			COMPLICATION:MEDICAL	4
COMPLICATION:MEDICAL	189	32		
			COMPLICATION:INFECTION	7
COMPLICATION:NEUROLOGICAL	30	18		
			COMPLICATION:POSTOP BLEED	1
COMPLICATION:POSTOP BLEED	45	14		

Figure 4.5 Default settings of the Audit Codes Summary Report

The default setting for the summary report is to list all of the audit data. Admission records are summarised by the incoming record's audit codes under a "Record Audit Code" column, with an "Audit Code" column displaying the total number of audit codes, and an "Admissions" column displaying the total number of admissions represented by those codes. For each summary group

the system's predictions of likely additional or alternative codes are also summarised using an "Additional/Alternative Codes" column for the code and an "Additions" column for the number of codes predicted.

Filter options are available to restrict the data viewed:

By the date range of the record using filter fields "Start Date" and "End Date";

picking from a multi-selection list restricting the incoming Record Audit Codes;

toggling between an AND / OR condition to join with choices of additional codes;

and using a multi-selection list restricting the choices of additional codes.

For example, a user wants to see all records for the year 2015 that involve types of complication coming from either the incoming record code *or* those suggested as additional/alternative codes:

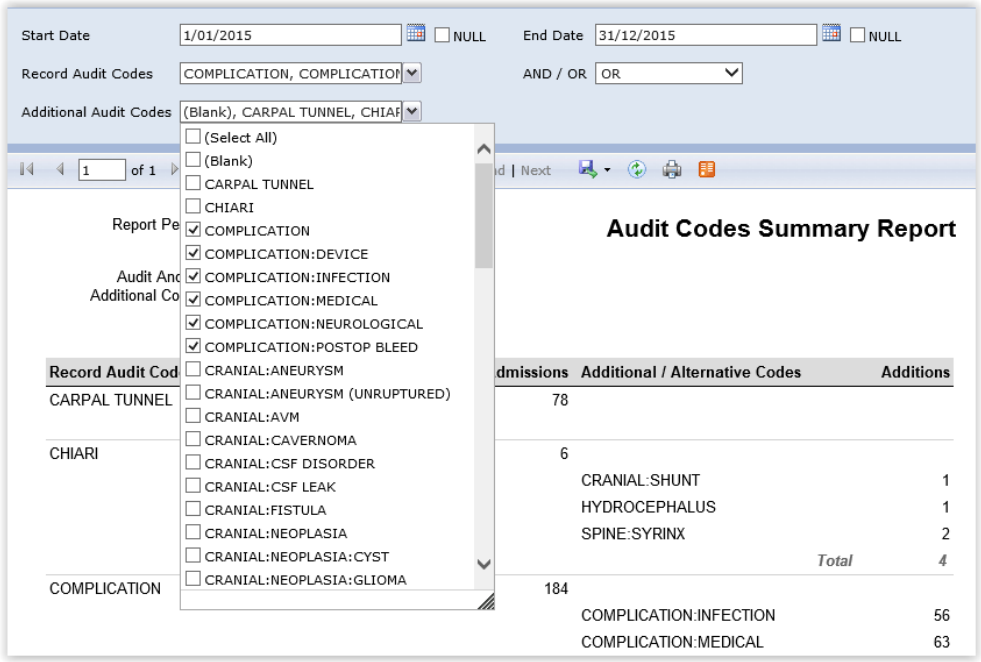
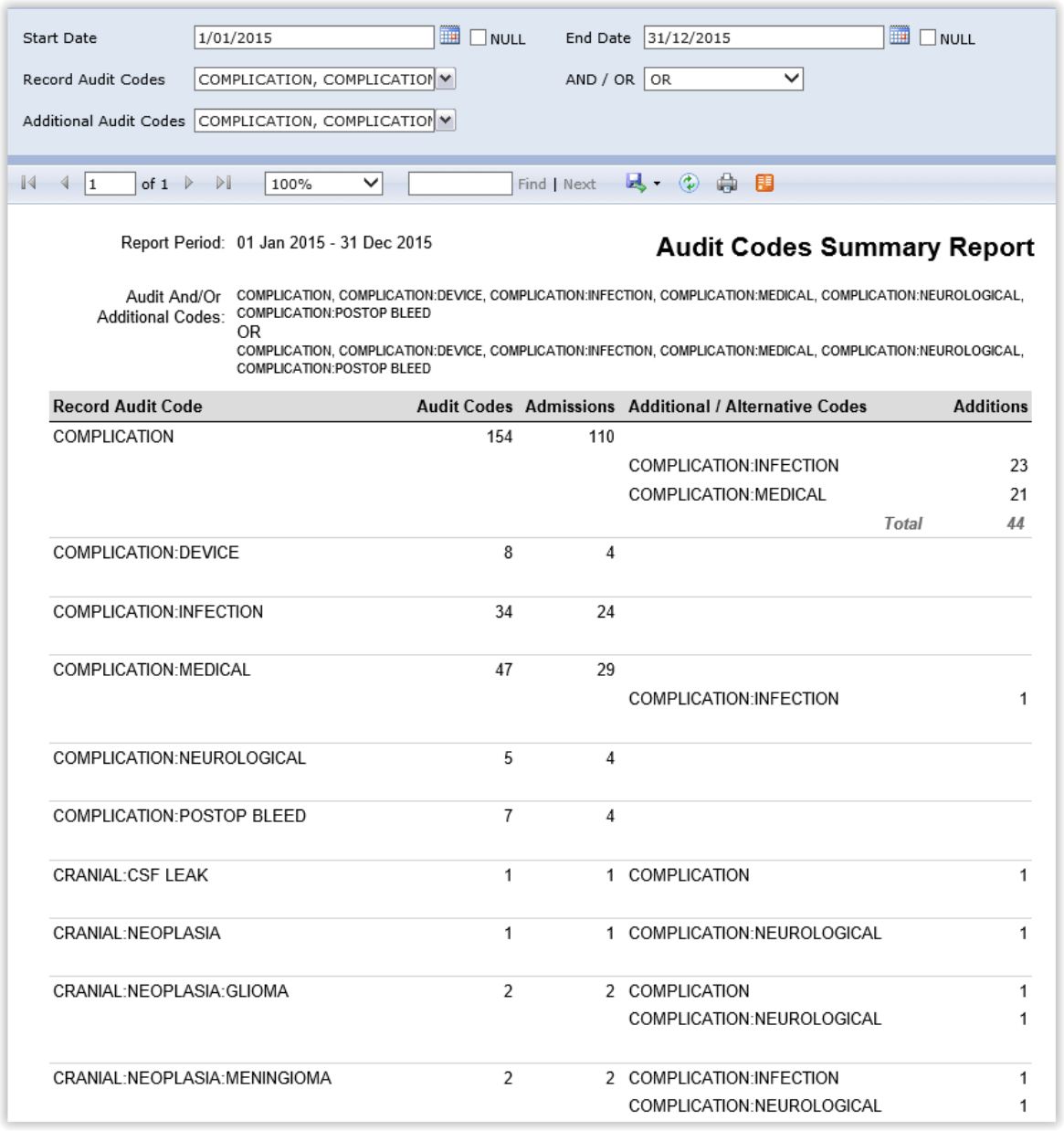


Figure 4.6 Setting up filters on the Audit Code Summary Report

By checking only complications-related record audit codes and additional codes but also selecting the OR option, then all complications-related records can be viewed – i.e. all complications-related incoming record audit codes combined with any additional codes which are also

complications-related, plus all remaining complications-related additional codes where the incoming code might not be complications-related:



Start Date 1/01/2015 NULL End Date 31/12/2015 NULL

Record Audit Codes COMPLICATION, COMPLICATION AND / OR OR

Additional Audit Codes COMPLICATION, COMPLICATION

1 of 1 100% Find | Next

Report Period: 01 Jan 2015 - 31 Dec 2015

Audit Codes Summary Report

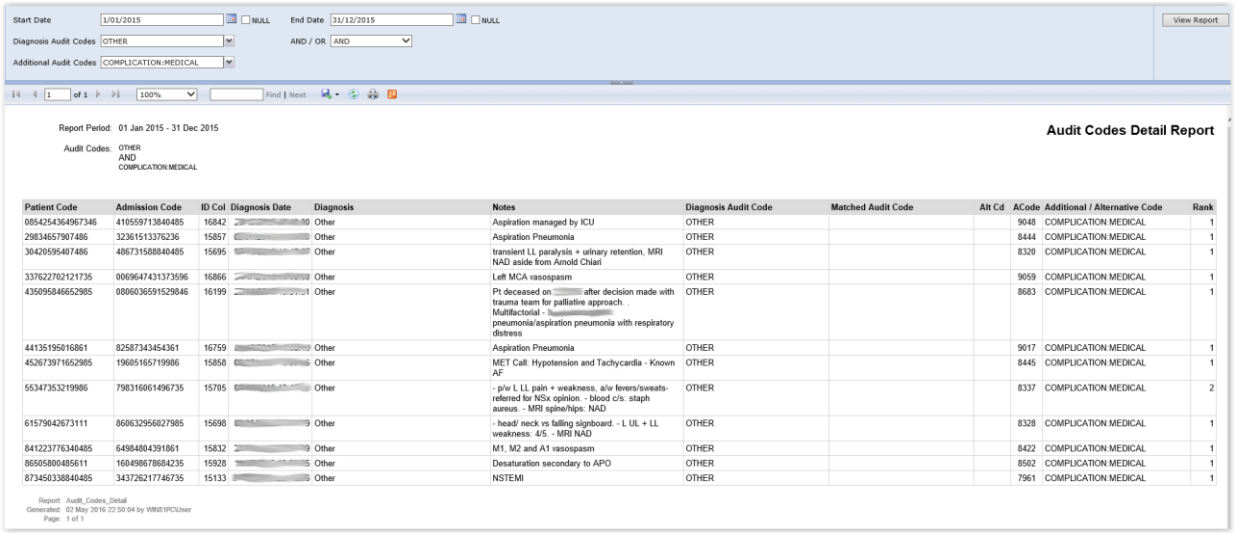
Audit And/Or Additional Codes: COMPLICATION, COMPLICATION:DEVICE, COMPLICATION:INFECTION, COMPLICATION:MEDICAL, COMPLICATION:NEUROLOGICAL, COMPLICATION:POSTOP BLEED
OR
COMPLICATION, COMPLICATION:DEVICE, COMPLICATION:INFECTION, COMPLICATION:MEDICAL, COMPLICATION:NEUROLOGICAL, COMPLICATION:POSTOP BLEED

Record Audit Code	Audit Codes	Admissions	Additional / Alternative Codes	Additions
COMPLICATION	154	110		
			COMPLICATION:INFECTION	23
			COMPLICATION:MEDICAL	21
			Total	*44*
COMPLICATION:DEVICE	8	4		
COMPLICATION:INFECTION	34	24		
COMPLICATION:MEDICAL	47	29		
			COMPLICATION:INFECTION	1
COMPLICATION:NEUROLOGICAL	5	4		
COMPLICATION:POSTOP BLEED	7	4		
CRANIAL:CSF LEAK	1	1	COMPLICATION	1
CRANIAL:NEOPLASIA	1	1	COMPLICATION:NEUROLOGICAL	1
CRANIAL:NEOPLASIA:GLIOMA	2	2	COMPLICATION	1
			COMPLICATION:NEUROLOGICAL	1
CRANIAL:NEOPLASIA:MENINGIOMA	2	2	COMPLICATION:INFECTION	1
			COMPLICATION:NEUROLOGICAL	1

Figure 4.7 The Audit Code Summary Report after setting filters

4.16.2 The Audit Codes Detail Report

Users can click on either of the Record Audit Code or Additional/Alternative Codes columns to obtain a detailed "drill-through" report which opens in a new browser window. For instance, clicking on the COMPLICATION:MEDICAL value in the Additional/Alternative Codes column linked to a Record Audit Code value of OTHER brings up the following detailed report:



Patient Code	Admission Code	ID Col	Diagnosis Date	Diagnosis	Notes	Diagnosis Audit Code	Matched Audit Code	Alt Cd	ACode	Additional / Alternative Code	Rank
0854254364967346	410559713840485	16842	[redacted]	Other	Aspiration managed by ICU	OTHER			9048	COMPLICATION:MEDICAL	1
29834657907486	32361513376236	15857	[redacted]	Other	Aspiration Pneumonia	OTHER			8444	COMPLICATION:MEDICAL	1
30420595407486	486731588840485	15695	[redacted]	Other	transient LL paralysis + urinary retention, MRI NAD aside from Arnold Chiari	OTHER			8320	COMPLICATION:MEDICAL	1
337622702121735	00696474313735 96	16866	[redacted]	Other	Left MCA vasospasm	OTHER			9059	COMPLICATION:MEDICAL	1
435095846652985	0806036591529846	16199	[redacted]	Other	Pt deceased on [redacted] after decision made with trauma team for palliative approach. . Multifactorial - [redacted] pneumonia/aspiration pneumonia with respiratory distress	OTHER			8683	COMPLICATION:MEDICAL	1
44135195016861	82587343454361	16759	[redacted]	Other	Aspiration Pneumonia	OTHER			9017	COMPLICATION:MEDICAL	1
452673971652985	19605165719986	15858	[redacted]	Other	MET Call: Hypotension and Tachycardia - Known AF	OTHER			8445	COMPLICATION:MEDICAL	1
553473532199 86	798316061496735	15705	[redacted]	Other	- p/w L LL pain + weakness, a/w fevers/sweats- referred for NSx opinion. - blood c/s: staph aureus. - MRI spine/hips: NAD	OTHER			8337	COMPLICATION:MEDICAL	2
61579042673111	860632956027985	15698	[redacted]	Other	- head/ neck vs falling signboard. - L UL + LL weakness: 4/5. - MRI NAD	OTHER			8328	COMPLICATION:MEDICAL	1
8412237776340485	64984804391861	15832	[redacted]	Other	M1, M2 and A1 vasospasm	OTHER			8422	COMPLICATION:MEDICAL	1
865058004856 11	160498678684235	15928	[redacted]	Other	Desaturation secondary to APO	OTHER			8502	COMPLICATION:MEDICAL	1
8734503388404 85	343726217746735	15133	[redacted]	Other	NSTEMI	OTHER			7961	COMPLICATION:MEDICAL	1

Figure 4.8 The Audit Code Detail Report

The filter settings and therefore the record selection conform to the filter choices made in the summary report, with the date range following the summary report but the diagnosis record and additional audit code choices reflecting the relevant values of the summary record that were clicked on. The detail report can also be run independently, in which case the filter options are interactive and the same as those available in the summary report.

The detail report shows the Patient and Admission record numbers, though in the data being used for the research and therefore also in the examples here, these are de-identified. The ID column is an internally assigned unique identifier for each incoming record, this is followed by the Diagnosis Date, here smudged to prevent identification. The Diagnosis and Notes are the next two fields, then comes the incoming record's Diagnosis Audit Code. Any match that could be made to this code by the system it is shown in the Matched Audit Code column, in this example the system tried to predict something more exact than OTHER, so no matches are found. The

ACode column is a unique identifier for the following Additional/Alternative Code, which in this case all are the alternative prediction of COMPLICATION:MEDICAL. The final field is the ranking given to the suggested code by the prediction process, these are in a range of one to seven for rule-based predictions, and from eight to fourteen for SVM-based predictions.

Where a match is found, the adjacent column Alt Cd indicates when the match was based on a valid alternative code, where these are deemed interchangeable due to similarity of data in the Notes. For instance, CRANIAL:TRAUMA:CONTUSIONS often uses the same terms as CRANIAL:TRAUMA:TBI, so there is no point in suggesting contusions as well as TBI (traumatic brain injury):

Notes	Diagnosis Audit Code	Matched Audit Code	Alt Cd	ACode	Additional / Alternative Code	Rank
L frontal SDH, SAH, contusions	CRANIAL:TRAUMA:TBI	CRANIAL:TRAUMA:CONTUSIONS	Yes	7749	CRANIAL:TRAUMA:SAH	2
				7750	CRANIAL:TRAUMA:SDH	3
Severe TBI - unsurvivable injury	CRANIAL:TRAUMA:TBI	CRANIAL:TRAUMA:TBI				
Death due to severe TBI	CRANIAL:TRAUMA:TBI	CRANIAL:TRAUMA:TBI		8033	COMPLICATION	2
post craniectomy and evacuation of right traumatic EDH, with trauma flap	CRANIAL:TRAUMA:TBI			8176	CRANIAL:TRAUMA:EDH	1
TBI with occ#, SDH and tSAH	CRANIAL:TRAUMA:TBI	CRANIAL:TRAUMA:TBI		8101	CRANIAL:TRAUMA:SAH	2
				8102	CRANIAL:TRAUMA:SDH	3
Severe TBI post MVA	CRANIAL:TRAUMA:TBI	CRANIAL:TRAUMA:TBI				
Rt convexity a/c SDH b/g epilepsy w falls	CRANIAL:TRAUMA:TBI			8425	CRANIAL:TRAUMA:SDH	1
				8426	EPILEPSY	2
left EDH, haemorrhagic contusions and skull #	CRANIAL:TRAUMA:TBI			8634	CRANIAL:TRAUMA:ICH	1
				8635	CRANIAL:TRAUMA:SKULL FRACTURE	2
				8636	CRANIAL:TRAUMA:EDH	3
IVH, Left temporal contusion	CRANIAL:TRAUMA:TBI	CRANIAL:TRAUMA:CONTUSIONS	Yes	7813	CRANIAL:TRAUMA:IVH	2
TBI w R tentorial acute SDH post fall	CRANIAL:TRAUMA:TBI	CRANIAL:TRAUMA:TBI		8424	CRANIAL:TRAUMA:SDH	2
Mild TBI - ped vs car	CRANIAL:TRAUMA:TBI	CRANIAL:TRAUMA:TBI				
right frontal contusion/TBI	CRANIAL:TRAUMA:TBI	CRANIAL:TRAUMA:CONTUSIONS	Yes			
R aSDH, non-survivable	CRANIAL:TRAUMA:TBI			8844	CRANIAL:TRAUMA:SDH	1
Left SAH and parafalcine SDH post AA, intoxicated	CRANIAL:TRAUMA:TBI			8427	CRANIAL:TRAUMA:SDH	1
				8428	CRANIAL:TRAUMA:SAH	2
Mild TBI w occipital # & bifrontal tSAH	CRANIAL:TRAUMA:TBI	CRANIAL:TRAUMA:TBI		8943	CRANIAL:TRAUMA:SKULL FRACTURE	2
				8944	CRANIAL:TRAUMA:SAH	3

Figure 4.9 View of Matched and Additional Codes on Audit Code Detail Report

Note that multiple additional codes can be suggested per incoming audit code.

There are two further drill-through options from the detail report - if the Admission Code is clicked on then the Detail report is loaded into a new browser window showing all of the records pertaining to that admission; and if the Patient Code is clicked then all admission records pertaining to that patient are likewise loaded as a detail report in a new browser window. These

details allow a fully informed conclusion to be drawn as to whether or not suggested codes are required, as the auditor can view all of the linked data and may consider that the incoming audit code of another record adequately covers the suggested code of the current record.

4.17 Conclusion

The chapter introduced the evaluation aims and objectives, which are to analyse the success of the audit code extraction process so that the best hybrid approach can be derived, yielding the most accurate system for discovering additional and alternative audit codes from the neurosurgical departments free text. Following this was a detailed explanation of how accuracy and coverage is tested, using standard terms of precision, recall and F-measure; including a description of how data must be prepared as training and testing data sets, and how this was implemented in this case.

The chapter then went on to describe how data was prepared for and processed through various machine learning algorithms in order to derive the most suitable, which was decided as being the Weka Sequential Minimal Optimization (SMO) Support Vector Machine, because it was the most accurate and because its data could be adapted for use within the application as an integrated machine learning component. Sections on evaluating predicted codes and reporting additional codes laid out the in more detail the assessment of the core functionality of the system which is to predict additional and alternative audit codes.

Following this discussion, the chapter went on to assess the predictive powers of rule-based versus SVM-based methods, and how these could be combined to produce more predictive opportunities; followed by a discussion on how to assess audit codes that are designed to replace the incoming code, and so cannot be verified in the normal way. The chapter discussed strategies for increasing the accuracy of its predictions by eliminating inaccurate and unrequired predictions; and described various challenges encountered during evaluation. The chapter concluded with a presentation of a proposed Audit Code Report, which is designed to enable an auditor to easily navigate the existing audit data and discover the predicted additional and alternative codes.

The input of the domain expert demonstrated that no matter how well the system identifies components in the notes that match incoming audit codes, and discovers additional or alternative audit codes, these can only be suggestions to bring to the attention of an auditor. The auditor needs to decide what to do with the additional information, which might be to use the additional suggestions unmodified; or to combine the suggestions into a new summary audit code; or to ignore them altogether. These sorts of high-level decisions cannot be automated; so the system must be understood as a semi-automated audit code extraction method.

Chapter 5
Discussion and Conclusion

5.1 Research Summary

Prior research has described that most electronic health records extensively use narrative text. Entering information as free-form text is the most natural and expressive way for clinicians to record the clinical encounter, however for analysis and re-use of this information, methods must be employed to extract and codify significant clinical information from clinical text. A variety of technologies have been used to extract coded clinical data via post-hoc text processing, including information extraction, natural language processing (NLP), data mining, and machine learning techniques (Meystre et al., 2008). Some researchers have concluded that the most effective results are obtained by combining technologies such as hand-crafted rules and machine learning (Farkas and Szarvas 2008).

Information Extraction (IE) is a process of retrieving specific targeted information from texts or speech and presenting them as explicit, codified data. A core component of IE is to categorise text into specific subject areas, a task called Named Entity Recognition (NER) (Meystre et al., 2008); Named Entities (NE) can then be utilised for code creation. Named Entity Recognition employs a sub-set of NLP techniques to pre-process text, then generally uses one or more rule-based, dictionary-based, or machine learning-based approaches (Krauthammer and Nenadic, 2004) to identify entities.

Systems developed to extract clinical information from the text of electronic health records range from many that have been focused on concise and often structured documents such as radiography (Huang et al., 2005) and echocardiogram results (Denny et al., 2009); to those that deal with documents having a much higher volume of often more grammatically normal text, such as discharge summaries (Melton and Hripcsak, 2005) and pathology reports (Coden et al., 2009). Some systems have been developed to extract a small component of the text, such as blood pressure results from physicians notes (Turchin et al., 2006), or family history from admission

notes (Friedlin and McDonald, 2006). If codes are being derived from these, they are usually standard codes such as UMLS or hospital systems billing codes, as a result there is an expanding resource of NLP systems designed to manage these tasks.

The research investigates the extraction of department-specific audit codes from the highly abbreviated, jargon heavy and domain-specific text used in a neurosurgical department of a major trauma hospital. The codes extracted should be additional or alternative codes after assessment of whether a linked incoming audit code is a suitable choice for the text. In addressing these challenges, research questions were proposed to investigate what intelligent techniques could be used for this task, and how to combine them to create a system that could be easily incorporated into the department's workflow and computer systems.

5.2 Addressing the Research Questions

The first research question of the thesis was:
"What intelligent techniques can be used to develop a semi-automated audit code extraction method?"

An exploration of the available clinical coding systems indicated that they were often unable to resolve the many neurological and neurosurgical specific terms found in the department's notes, and where they did find a term it was often not a straightforward match to a neurosurgical meaning, with the requirement to filter out many irrelevant interpretations. These systems perform NLP processing over the entire text, and consequently produced unwanted and irrelevant output compared with the targeted Information Extraction that analysis indicated was appropriate for the neurosurgical department. Finally, these systems were found to be quite slow to run due to the breadth of their coverage, and usually worked only as a stand-alone process - integrating them into existing software was therefore problematic.

After the assessment of existing applications, the conclusion reached was that a custom approach needed to be developed. Literature reviews of solutions for similar problems suggested that an

optimal approach would be a rule and dictionary based system augmented by machine learning components. The direction taken therefore was to verify this approach by the construction of a computer-based design which should be evaluated via a working software implementation of the design. In the terminology of Design Science (as described by Hevner et al. (2004)), the design is a method which is instantiated as software.

The textual data processed by the system comes from the admission records of the neurosurgical department, which consist of a note and an attached diagnosis. Processing takes place in a series of steps – starting with a preprocessing stage designed to identify text of interest, which means that some words are filtered out, others are corrected or disambiguated, and if possible distinct sentences are identified – on the premise that a sentence is more likely to focus on one subject. The appropriate audit code is associated with the incoming diagnosis; this code becomes the default class label for learning and evaluating a Support Vector Machine algorithm on the collected sentences, which takes place as a one-off (but repeatable) step – resulting in data that contains the output of the SVM learning and that can be then incorporated into the system.

The steps outlined so far culminate in an off-line machine learning step, in order to be able to incorporate the data coming from that step. The evaluation of machine learning algorithms and the rationale for choosing a Sequential Minimal Optimization (SMO) Support Vector Machine (SVM) are described in chapter three. Once the machine learning has been performed the evaluation of the text can be completed through the remaining stages, which includes input from the ML system. Additionally, the system is prepared to work in a real-time manner, as the remaining steps are capable of being implemented as a real-time sequence: Receiving the input of an individual record, processing it through the text preparation step and the following rule-based audit code identification step, then handing over the extracted sentences to a real-time implementation of the machine learning-based code identification component.

The rule-based text processing component looks for specific words and phrases that audit codes can be assigned to, while all remaining words are identified where possible as belonging to either the medical domain or to descriptions of the cause of admission. Sentences are tagged with as

many audit codes as can be found, which typically are no more than three codes. If the incoming audit code is one of those that are amenable to input from the machine learning component then the sentences are also passed through that, so finally each sentence may have additional audit codes identified by the ML component. All codes are then ranked for relevance and conformity to the incoming audit code, and only useful additional or alternative codes are identified to form the output of the process.

Verification of the system, described in chapter four, was carried out by evaluating the accuracy of its rule-based and machine learning-based extraction of audit codes, according to standard measures of precision, recall, and F-measure. The most suitable combinations of the two techniques were likewise identified as those where the combination resulted in a higher F-measure. Further refinement of the accuracy of the system is achieved by eliminating inaccurate or un-useful audit code predictions according to their consistency with the accompanying diagnosis of the admission record, this logic is incorporated into the final audit code ranking step of the working system.

The second research question of the thesis was:
"How can a code extraction solution be designed to be applicable to both the audit process and the initial data entry process?"

To produce a light-weight and responsive application, and also enabling integration into the department's system, the software was written as a series of SQL functions, which chain together to perform the various tasks required. The presentation of the identified additional codes resulting from the system is via a report or dashboard, to be used by the auditors of the neurosurgical department, but the design of the system allows for possible incorporation into the neurosurgical department's data-entry software, as the system can work entirely in real-time once machine learning on the text has been performed and its output incorporated.

5.3 Results Overview

Evaluation of the results of the main components of the system were performed using standard terms of precision, recall and F-measure; it was found that overall the rule-based system on its own out-performed the any machine learning-based systems evaluated, with an F-measure of 0.749 for the rule-based versus 0.672 for the best ML-based system, which was a Weka Sequential Minimal Optimization (SMO) Support Vector Machine. Nevertheless, the SVM produced a better predictive outcome for a third of the audit codes, and combining these two methods for those codes where the overall F-measure was increased produced a marginally better overall F-measure of 0.751. The accuracy of the predictions can be improved by eliminating inaccurate and unrequired predictions, which is a separate rule-based step.

The author could not find any published research on equivalent NLP applications for processing neurosurgical notes into in-house audit codes, so the scoring of the application cannot be compared relative to an equivalent system.

Because the output of the system is additional or alternative audit codes compared with the incoming audit code of a Note, the final accuracy of the system cannot be evaluated by comparing the accuracy of the final predictions against the incoming audit code – the system is trying to find everything *but* the incoming code – so this evaluation has to be performed by a neurosurgical department expert.

5.4 Contribution

An audit code extraction method was designed, that matches neurosurgical-specific audit codes to the highly abbreviated jargon found in clinical notes of the neurosurgical department of a major trauma hospital, using a hybrid of rule and dictionary-based information extraction techniques, augmented by a machine learning component. The method was instantiated as a working computer program, which accepts an input of a Note and accompanying audit code, and produces an output of a report of matched audit codes, and suggested additional or alternative audit codes. The output was able to be verified for accuracy using standard evaluation metrics, and as the

software and output can be evaluated in situ in the neurosurgical department, it will be able to be further verified by the department's auditors.

The software will integrate into the SQL back-end of the department's systems; and is capable of being extended to function in the front-end for diagnostic code-matching during data entry. Although the rules encapsulated into the system are specific, the design of the various components allows them to be applied to similar situations.

An innovative aspect of the application is the creation of a function that uses the data output from an off-line machine learning process to incorporate ML predictions into the programme. The resulting application is an innovative targeted information extraction system which incorporates machine learning that is lightweight enough to be able to process in real-time if required, and to be easily maintained and extended.

In summary the thesis's contribution is:

- A novel computer-based method that is designed to precisely and rapidly match neurosurgical audit and diagnostic codes from the free text notes of the neurosurgical department.
- The construction of a software tool to instantiate the method using technology that will integrate into the current systems of the neurosurgical department.
- The design includes the capacity for continuous improvement by a system expert refining the rules and medical ontologies used, and by further training of machine learning components.
- To design also enables application to other similar areas where a speciality uses its own codes matched to abbreviated free text notes.

5.5 Limitations

Resource constraints in the neurosurgical department meant that a comprehensive evaluation of the final results was unable to be conducted. Consequently, conclusions about the system's

efficacy are largely on the basis of its accuracy in matching the incoming audit category, with a limited confirmation by a domain expert of the system's reported additional and alternative codes.

Although other clinical records from outside of the neurosurgical department exist, they were not available to this research. In any case the neurosurgical department auditor stated that the external systems do not contain information that could be harnessed to refine this system, which is highly specific to suggesting additional audit codes from the existing departmental notes. It is possible though, that had other systems been available and capable of being integrated into the text analysis, further likely audit codes could have emerged.

The current design of the system is not immediately generalizable, as the algorithms used for identifying key words are embedded into the programme. The programme structure could be used in another context, but the specific embedded rules would need to be re-written. Nevertheless, the modular design of the programme, and its accessible SQL Server-based technology, allows for quite easy adaptation to another similar problem area.

5.6 Future Research

The most practical next step would be to install the programme into the neurosurgical department's systems and start to use it, and to refine the rules based on an evaluation by the department; proper evaluation was unfortunately not possible due to personnel resourcing issues in the department during the latter stages of development of the programme. Linked to this and also dependent on departmental evaluation would be to improve the filtering of irrelevant codes, as any refinement of the output by pruning unwanted data improves the overall accuracy.

There is scope for improving the way in which the rule-based and machine learning-based systems are combined. Combining them with a logical OR for the prediction is a rather simple approach, it should be possible to devise better algorithms for combining the best of each. The complexity in the approach of using only certain combinations is that it depends on accepting only certain

predictions (like COMPLICATION), there are no doubt better ways of testing the current combination without having to know what the prediction is, in relation to a fixed list.

There is an interesting possibility that the meta data generated by the rule-based system could be combined with the text to provide a greater number of patterns for the machine learning system to learn on. For instance, there are many words that are classified by the rule-based system as anatomical, such as "anatomical:brain", "anatomical:skull", and "anatomical:spine". Currently the rule-based system does not use this data but since these classifications tie together many more complex words, some of which appear very infrequently, it's possible that adding these classifications to the text, and even also the audit codes deduced by the rules-based system, would assist the ML process to be more accurate.

The use of the output of the ML support vector machine to provide a table of word weightings proved to be very useful, it meant that this data could be made use of by the application directly, rather than having to pass processing back to the ML software. There is opportunity to refine the way the ML data is used, especially when it comes to eliminating obvious mistakes by the ML system. The next most accurate ML algorithm was the Naïve Bayes process, and its output is also amenable to incorporation into the application, so it would be a good to be able to test the result of combining these two ML systems through this architecture.

Other machine learning processes could also be evaluated, and other ML software – this research used the Weka data mining software but it would be interesting to explore libraries written in R or Python. R is especially an intriguing option as with the advent of SQL Server 2016 it is now possible to embed R functionality into a SQL Server programme. A routine written in R was used to convert the text output of the Weka SMO machine into SQL data.

Using programmatically-defined rules was most efficient and maintainable approach to implementing the system, even the smaller dictionaries employed by the system are embedded into the SQL code, with only the Domain Concept dictionary being table-based. This approach means that it is a relatively simple matter to understand and alter the system, by reading and

writing SQL code. However, all of the dictionaries could be separated out into tables, and more significantly, the techniques used to match audit codes and key words could be separated from the actual audit codes and key words being tested. To do this, a dictionary could be constructed to hold audit codes with accompanying key words and their interrelationships, and these could be cycled through programmatically. Taking this approach would require an additional component to allow for the maintenance of the rules dictionary, with a goal for it to be as understandable and maintainable as the current code-based approach. In conclusion, an opportunity exists for future research into the system design, to improve its generalizability.

References

Agah, A., 2013. Medical Applications of Artificial Intelligence. CRC Press.

Aronson, A.R., 2001. Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. Proc. AMIA Symp. 17–21.

Aronson, A.R., Bodenreider, O., Demner-Fushman, D., Fung, K.W., Lee, V.K., Mork, J.G., Névéol, A., Peters, L., Rogers, W.J., 2007. From Indexing the Biomedical Literature to Coding Clinical Text: Experience with MTI and Machine Learning Approaches, in: Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing, BioNLP '07. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 105–112.

Aronson, A.R., Lang, F.-M., 2010. An overview of MetaMap: historical perspective and recent advances. J. Am. Med. Inform. Assoc. 17, 229–236. doi:10.1136/jamia.2009.002733

Aronson, A.R., Rindflesch, T.C., 1997. Query expansion using the UMLS Metathesaurus. Proc. AMIA Annu. Fall Symp. 485–489.

Ash, J.S., Berg, M., Coiera, E., 2004. Some Unintended Consequences of Information Technology in Health Care: The Nature of Patient Care Information System-related Errors. J. Am. Med. Inform. Assoc. JAMIA 11, 104–112. doi:10.1197/jamia.M1471

Bashyam, V., Divita, G., Bennett, D.B., Taira, R.K., Browne, A.C., 2007. A Normalized Lexical Lookup Approach to Identifying UMLS Concepts in Free Text [WWW Document]. URL http://search.informit.com.au/documentSummary;dn=782137008281707;res=IELHEA (accessed 4.7.15).

Bellazzi, R., Zupan, B., 2008. Predictive data mining in clinical medicine: Current issues and guidelines. Int. J. Med. Inf. 77, 81–97. doi:10.1016/j.ijmedinf.2006.11.006

Chapman, W.W., Christensen, L.M., Wagner, M.M., Haug, P.J., Ivanov, O., Dowling, J.N., Olszewski, R.T., 2005. Classifying free-text triage chief complaints into syndromic categories with natural language processing. Artif. Intell. Med. 33, 31–40. doi:10.1016/j.artmed.2004.04.001

Christensen, L.M., Haug, P.J., Fiszman, M., 2002. MPLUS: A Probabilistic Medical Language Understanding System, in: Proceedings of the ACL-02 Workshop on Natural Language Processing in the Biomedical Domain - Volume 3, BioMed '02. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 29–36. doi:10.3115/1118149.1118154

Clark, C., Good, K., Jezierny, L., Macpherson, M., Wilson, B., Chajewska, U., 2008. Identifying Smokers with a Medical Extraction System. J. Am. Med. Inform. Assoc. 15, 36–39. doi:10.1197/jamia.M2442

Coden, A., Savova, G., Sominsky, I., Tanenblatt, M., Masanz, J., Schuler, K., Cooper, J., Guan, W., de Groen, P.C., 2009. Automatically extracting cancer disease characteristics from pathology reports into a Disease Knowledge Representation Model. J. Biomed. Inform., Biomedical Natural Language Processing 42, 937–949. doi:10.1016/j.jbi.2008.12.005

Crammer, K., Dredze, M., Ganchev, K., Talukdar, P., Carroll, S., 2007. Automatic Code Assignment to Medical Text. Presented at the Biological, translational, and clinical language processing, Association for Computational Linguistics, pp. 129–136.

Cunningham, H., 2005. Information Extraction, Automatic, in: Encyclopedia of Language and Linguistics. pp. 665–677.
Day, S., Christensen, L.M., Dalto, J., Haug, P., 2007. Identification of Trauma Patients at a Level 1 Trauma Center Utilizing Natural Language Processing. J. Trauma Nurs. 14, 79–83.
Demner-Fushman, D., Chapman, W.W., McDonald, C.J., 2009. What can natural language processing do for clinical decision support? J. Biomed. Inform., Biomedical Natural Language Processing 42, 760–772. doi:10.1016/j.jbi.2009.08.007
Denny, J.C., Miller, R.A., Waitman, L.R., Arrieta, M.A., Peterson, J.F., 2009. Identifying QT prolongation from ECG impressions using a general-purpose Natural Language Processor. Int. J. Med. Inf., MedInfo 2007 78, Supplement 1, S34–S42. doi:10.1016/j.ijmedinf.2008.09.001
Doan, S., Conway, M., Phuong, T.M., Ohno-Machado, L., 2014. Natural language processing in biomedicine: a unified system architecture overview. Methods Mol. Biol. Clifton NJ 1168, 275–294. doi:10.1007/978-1-4939-0847-9_16
Farkas, R., Szarvas, G., 2008. Automatic construction of rule-based ICD-9-CM coding systems. BMC Bioinformatics 9, S10. doi:10.1186/1471-2105-9-S3-S10
Fisk, J.M., Mutalik, P., Levin, F.W., Erdos, J., Taylor, C., Nadkarni, P., 2003. Integrating Query of Relational and Textual Data in Clinical Databases: A Case Study. J. Am. Med. Inform. Assoc. 10, 21–38. doi:10.1197/jamia.M1133
Fiszman, M., Chapman, W.W., Aronsky, D., Evans, R.S., Haug, P.J., 2000. Automatic Detection of Acute Bacterial Pneumonia from Chest X-ray Reports. J. Am. Med. Inform. Assoc. 7, 593–604. doi:10.1136/jamia.2000.0070593
Friedlin, J., McDonald, C.J., 2006. A Natural Language Processing System to Extract and Code Concepts Relating to Congestive Heart Failure from Chest Radiology Reports. AMIA. Annu. Symp. Proc. 2006, 269–273.
Friedman, C., Alderson, P.O., Austin, J.H.M., Cimino, J.J., Johnson, S.B., 1994. A General Natural-language Text Processor for Clinical Radiology. J. Am. Med. Inform. Assoc. 1, 161–174. doi:10.1136/jamia.1994.95236146
Fung, K.W., Bodenreider, O., 2012. Knowledge Representation and Ontologies, in: Richesson, R.L., Andrews, J.E. (Eds.), Clinical Research Informatics, Health Informatics. Springer London, pp. 255–275.
Gaizauskas, R., Harkema, H., Hepple, M., Setzer, A., 2006. Task-Oriented Extraction of Temporal Information: The Case of Clinical Narratives, in: Thirteenth International Symposium on Temporal Representation and Reasoning, 2006. TIME 2006. Presented at the Thirteenth International Symposium on Temporal Representation and Reasoning, 2006. TIME 2006, pp. 188–195. doi:10.1109/TIME.2006.27
GATE [WWW Document], n.d. URL https://gate.ac.uk/commercial.html (accessed 5.12.15).
Gold, S., Elhadad, N., Zhu, X., Cimino, J.J., Hripcsak, G., 2008. Extracting Structured Medication Event Information from Discharge Summaries. AMIA. Annu. Symp. Proc. 2008, 237–241.
Goldstein, I., Arzumtsyan, A., Uzuner, Ö., 2007. Three Approaches to Automatic Assignment of ICD-9-CM Codes to Radiology Reports. AMIA. Annu. Symp. Proc. 2007, 279–283.
Greenhalgh, T., Potts, H.W.W., Wong, G., Bark, P., Swinglehurst, D., 2009. Tensions and Paradoxes in Electronic Patient Record Research: A Systematic Literature Review Using

the Meta-narrative Method. Milbank Q. 87, 729–788. doi:10.1111/j.1468-0009.2009.00578.x
Grishman, R., 1997. Information extraction: Techniques and challenges, in: Pazienza, M.T. (Ed.), Information Extraction A Multidisciplinary Approach to an Emerging Information Technology, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 10–27.
Grishman, R., Sundheim, B., 1996. Message Understanding Conference-6: A Brief History., in: COLING. pp. 466–471.
Gruber, T.R., 1993. A translation approach to portable ontology specifications. Knowl. Acquis. 5, 199–220. doi:10.1006/knac.1993.1008
Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene Selection for Cancer Classification using Support Vector Machines. Mach. Learn. 46, 389–422. doi:10.1023/A:1012487302797
Haug, P.J., Christensen, L., Gundersen, M., Clemons, B., Koehler, S., Bauer, K., 1997. A natural language parsing system for encoding admitting diagnoses. Proc. AMIA Annu. Fall Symp. 814–818.
Haug, P.J., Koehler, S., Lau, L.M., Wang, P., Rocha, R., Huff, S.M., 1995. Experience with a mixed semantic/syntactic parser. Proc. Annu. Symp. Comput. Appl. Med. Care 284–288.
Hevner, A.R., March, S.T., Park, J., Ram, S., 2004. Design Science in Information Systems Research. MIS Q 28, 75–105.
Hripcsak, G., Kuperman, G.J., Friedman, C., 1998. Extracting findings from narrative reports: software transferability and sources of physician disagreement. Methods Inf Med 37, 1–7.
Huang, Y., Lowe, H.J., Klein, D., Cucina, R.J., 2005. Improved Identification of Noun Phrases in Clinical Radiology Reports Using a High-Performance Statistical Natural Language Parser Augmented with the UMLS Specialist Lexicon. J. Am. Med. Inform. Assoc. JAMIA 12, 275–285. doi:10.1197/jamia.M1695
Huang, Y., McCullagh, P., Black, N., Harper, R., 2007. Feature selection and classification model construction on type 2 diabetic patients' data. Artif. Intell. Med. 41, 251–262. doi:10.1016/j.artmed.2007.07.002
Iavindrasana, J., Cohen, G., Depeursinge, A., Müller, H., Meyer, R., Geissbuhler, A., 2009. Clinical data mining: a review. Yearb. Med. Inform. 121–133.
Imler, T.D., Morea, J., Kahi, C., Imperiale, T.F., 2013. NATURAL LANGUAGE PROCESSING ACCURATELY CATEGORIZES FINDINGS FROM COLONOSCOPY AND PATHOLOGY REPORTS. Clin. Gastroenterol. Hepatol. Off. Clin. Pract. J. Am. Gastroenterol. Assoc. 11, 689–694. doi:10.1016/j.cgh.2012.11.035
Ivanović, M., Budimac, Z., 2014. An overview of ontologies and data resources in medical domains. Expert Syst. Appl. 41, 5158–5166. doi:10.1016/j.eswa.2014.02.045
Kaplan, D.M., 2007. Clear writing, clear thinking and the disappearing art of the problem list. J. Hosp. Med. 2, 199–202. doi:10.1002/jhm.242
Kohane, I., Uzuner, Ö., 2008. No Structure Before Its Time. J. Am. Med. Inform. Assoc. JAMIA 15, 708. doi:10.1197/jamia.M2781
Krauthammer, M., Nenadic, G., 2004. Term identification in the biomedical literature. J. Biomed. Inform., Named Entity Recognition in Biomedicine 37, 512–526. doi:10.1016/j.jbi.2004.08.004

Lehnert, W., SODERLAND, S., ARONOW, D., FENG, F., SHMUELI, A., 1995. INDUCTIVE TEXT CLASSIFICATION FOR MEDICAL APPLICATIONS. Comput. Sci. Dep. Fac. Publ. Ser. 7.

Leo. Breiman, 1993. Classification and regression trees. Chapman & Hall/CRC, Boca Raton, Fla.

Lewis, D.D., 1998. Naive (Bayes) at forty: The independence assumption in information retrieval, in: Nédellec, C., Rouveirol, C. (Eds.), Machine Learning: ECML-98, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 4–15.

Liu, H., Lussier, Y.A., Friedman, C., 2001. Disambiguating Ambiguous Biomedical Terms in Biomedical Narrative Text: An Unsupervised Method. J. Biomed. Inform. 34, 249–261. doi:10.1006/jbin.2001.1023

Long, W., 2005. Extracting Diagnoses from Discharge Summaries. AMIA. Annu. Symp. Proc. 2005, 470–474.

Luke Butt, G.Z., 2013. Classification of cancer-related death certificates using machine learning. Australas. Med. J. 6, 292–9. doi:10.4066/AMJ.2013.1654

McNaught, J., Black, W., 2006. Information extraction. Text Min. Biol. Biomed. 143–179.

Melton, G.B., Hripcsak, G., 2005. Automated Detection of Adverse Events Using Natural Language Processing of Discharge Summaries. J. Am. Med. Inform. Assoc. JAMIA 12, 448–457. doi:10.1197/jamia.M1794

Meystre, S., Haug, P.J., 2006a. Natural language processing to extract medical problems from electronic clinical documents: Performance evaluation. J. Biomed. Inform. 39, 589–599. doi:10.1016/j.jbi.2005.11.004

Meystre, S., Haug, P.J., 2006b. Natural language processing to extract medical problems from electronic clinical documents: Performance evaluation. J. Biomed. Inform. 39, 589–599. doi:10.1016/j.jbi.2005.11.004

Meystre, S., Savova, G., Kipper-Schuler, K., Hurdle, J., 2008. Extracting information from textual documents in the electronic health record: a review of recent research. Yearb. Med. Inform. 128–144.

Nadkarni, P.M., Ohno-Machado, L., Chapman, W.W., 2011. Natural language processing: an introduction. J. Am. Med. Inform. Assoc. JAMIA 18, 544–551. doi:10.1136/amiajnl-2011-000464

Napolitano, G., Fox, C., Middleton, R., Connolly, D., 2010. Pattern-based information extraction from pathology reports for cancer registration. Cancer Causes Control 21, 1887–94. doi:http://dx.doi.org.ezproxy.lib.monash.edu.au/10.1007/s10552-010-9616-4

Pakhomov, S.V.S., Buntrock, J.D., Chute, C.G., 2006. Automating the Assignment of Diagnosis Codes to Patient Encounters Using Example-based and Machine Learning Techniques. J. Am. Med. Inform. Assoc. JAMIA 13, 516–525. doi:10.1197/jamia.M2077

Pestian, J.P., Brew, C., Matykiewicz, P., Hovermale, D.J., Johnson, N., Cohen, K.B., Duch, W., 2007. A Shared Task Involving Multi-label Classification of Clinical Free Text, in: Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing, BioNLP '07. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 97–104.

Platt, J.C., 1999. Fast Training of Support Vector Machines Using Sequential Minimal Optimization, in: Schölkopf, B., Burges, C.J.C., Smola, A.J. (Eds.), Advances in Kernel Methods : Support Vector Learning. MIT Press, Cambridge, MA, USA, pp. 185–208.

Price, M., Singer, A., Kim, J., 2013. Adopting electronic medical records Are they just electronic paper records? Can. Fam. Physician 59, e322–e329.
Principles for Best Practice in Clinical Audit, 2002. . Radcliffe Publishing.
Quinlan, J.R., 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann.
Ranum, D.L., 1989. Knowledge-based understanding of radiology text. Comput. Methods Programs Biomed. 30, 209–215. doi:10.1016/0169-2607(89)90073-4
Rector, A.L., 1999. Clinical terminology: why is it so hard? Methods Inf. Med. 38, 239–252. doi:10.1267/METH99040239
Roberts, A., 2012. Clinical Information Extraction: Lowering the Barrier (Thesis). University of Sheffield.
Rosenbloom, S.T., Denny, J.C., Xu, H., Lorenzi, N., Stead, W.W., Johnson, K.B., 2011. Data from clinical notes: a perspective on the tension between structure and flexible documentation. J. Am. Med. Inform. Assoc. JAMIA 18, 181–186. doi:10.1136/jamia.2010.007237
Rosenbloom, S.T., Miller, R.A., Johnson, K.B., Elkin, P.L., Brown, S.H., 2006. Interface Terminologies. J. Am. Med. Inform. Assoc. 13, 277–288. doi:10.1197/jamia.M1957
Sager, N., Lyman, M., Bucknall, C., Nhan, N., Tick, L.J., 1994. Natural language processing and the representation of clinical data. J. Am. Med. Inform. Assoc. 1, 142–160.
Savova, G.K., Masanz, J.J., Ogren, P.V., Zheng, J., Sohn, S., Kipper-Schuler, K.C., Chute, C.G., 2010. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. J. Am. Med. Inform. Assoc. JAMIA 17, 507–513. doi:10.1136/jamia.2009.001560
Schön, D.A., 1991. The reflective practitioner : how professionals think in action. Arena, Aldershot, England.
SNOMED CT [WWW Document], n.d. URL http://www.ihtsdo.org/snomed-ct/what-is-snomed-ct/how-does-snomed-ct-work (accessed 4.23.15).
Spasić, I., Livsey, J., Keane, J.A., Nenadić, G., 2014. Text mining of cancer-related information: review of current status and future directions. Int. J. Med. Inf. 83, 605–623. doi:10.1016/j.ijmedinf.2014.06.009
Spat, S., B, C., I, R., C, G., H, L., G, S., P, B., 2007. Enhanced information retrieval from narrative German-language clinical text documents using automated document classification. Stud. Health Technol. Inform. 136, 473–478.
Spyns, P., 1996. Natural language processing in medicine: an overview. Methods Inf. Med. 35, 285–301.
Stanfill, M.H., Williams, M., Fenton, S.H., Jenders, R.A., Hersh, W.R., 2010. A systematic literature review of automated clinical coding and classification systems. J. Am. Med. Inform. Assoc. JAMIA 17, 646–651. doi:10.1136/jamia.2009.001024
Stanfill, M.H., Williams, M., Fenton, S.H., Jenders, R.A., Hersh, W.R., 2010. A systematic literature review of automated clinical coding and classification systems. J. Am. Med. Inform. Assoc. 17, 646–651. doi:10.1136/jamia.2009.001024
Suominen, H., Ginter, F., Pyysalo, S., Airola, A., Pahikkala, T., Salanter, S., Salakoski, T., 2008. Machine learning to automate the assignment of diagnosis codes to free-text radiology reports: a method description, in: Proceedings of the ICML/UAI/COLT Workshop on Machine Learning for Health-Care Applications.

Turchin, A., Kolatkar, N.S., Grant, R.W., Makhni, E.C., Pendergrass, M.L., Einbinder, J.S., 2006. Using Regular Expressions to Abstract Blood Pressure and Treatment Intensification Information from the Text of Physician Notes. J. Am. Med. Inform. Assoc. JAMIA 13, 691–695. doi:10.1197/jamia.M2078

Turney, P.D., Pantel, P., 2010. From Frequency to Meaning: Vector Space Models of Semantics. J Artif Int Res 37, 141–188.

UMLS Quick Start Guide [WWW Document], n.d. URL http://www.nlm.nih.gov/research/umls/quickstart.html (accessed 4.23.15).

Uzuner, Ö., South, B.R., Shen, S., DuVall, S.L., 2011. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. J. Am. Med. Inform. Assoc. 18, 552–556. doi:10.1136/amiajnl-2011-000203

van Ginneken, A.M., 2002. The computerized patient record: balancing effort and benefit. Int. J. Med. Inf. 65, 97–119. doi:10.1016/S1386-5056(02)00007-2

Walsh, S.H., 2004. The clinician's perspective on electronic health records and how they can affect patient care. BMJ 328, 1184–1187.

Wang, Y., Patrick, J., 2009. Cascading Classifiers for Named Entity Recognition in Clinical Notes, in: Proceedings of the Workshop on Biomedical Information Extraction, WBIE '09. Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 42–49.

WHO | International Classification of Diseases (ICD) [WWW Document], n.d. . WHO. URL http://www.who.int/classifications/icd/en/ (accessed 2.19.15).

Witten, I.H., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques, Second Edition. Morgan Kaufmann.

Witten, I.H., Frank, E., Hall, M.A., 2011. Chapter 5 - Credibility: Evaluating What's Been Learned, in: Data Mining: Practical Machine Learning Tools and Techniques (Third Edition), The Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, Boston, pp. 147–187.

Wright, A., Chen, E.S., Maloney, F.L., 2010. An automated technique for identifying associations between medications, laboratory results and problems. J. Biomed. Inform. 43, 891–901. doi:10.1016/j.jbi.2010.09.009

Wright, A., Pang, J., Feblowitz, J.C., Maloney, F.L., Wilcox, A.R., Ramelson, H.Z., Schneider, L.I., Bates, D.W., 2011. A method and knowledge base for automated inference of patient problems from structured data in an electronic medical record. J. Am. Med. Inform. Assoc. 18, 859–867. doi:10.1136/amiajnl-2011-000121

Xu, H., Stetson, P.D., Friedman, C., 2009. Methods for Building Sense Inventories of Abbreviations in Clinical Notes. J. Am. Med. Inform. Assoc. 16, 103–108. doi:10.1197/jamia.M2927

Zuccon, G., Wagholikar, A.S., Nguyen, A.N., Butt, L., Chu, K., Martin, S., Greenslade, J., 2013. Automatic classification of free-text radiology reports to identify limb fractures using machine learning and the SNOMED CT ontology, in: AMIA Summits on Translational Science Proceedings. Presented at the AMIA Summits on Translational Science 2013, American Medical Informatics Association, San Francisco, California, pp. 300–304.

Appendix A Prediction Evaluation Comparisons

DiagnosisAuditCode	Class Total	Rule Based Matched	Rule Based Predicted	Rule Based Precision	Rule Based Recall	Rule Based FScore	SVM Based Matched	SVM Based Predicted	SVM Based Precision	SVM Based Recall	SVM Based FScore	Combined Matched	Combined Predicted	Combined Precision	Combined Recall	Combined FScore	Combined-Rule based Difference	Hybrid FScore Source	Hybrid Matched	Hybrid Predicted	Hybrid Precision	Hybrid Recall	Hybrid FScore
CARPAL TUNNEL	81	81	85	0.953	1.000	0.976	65	163	0.399	0.802	0.533	81	180	0.450	1.000	0.621	-0.355	Rule Based	81	85	0.953	1.000	0.976
CHIARI	8	5	8	0.625	0.625	0.625	5	5	1.000	0.625	0.769	6	9	0.667	0.750	0.706	0.081	Rule Based	5	8	0.625	0.625	0.625
COMPLICATION	465	144	171	0.842	0.310	0.453	170	224	0.759	0.366	0.494	223	300	0.743	0.480	0.583	0.130	Combined	223	300	0.743	0.480	0.583
COMPLICATION:DEVICE	11	9	9	1.000	0.818	0.900	8	12	0.667	0.727	0.696	10	14	0.714	0.909	0.800	-0.100	Rule Based	9	9	1.000	0.818	0.900
COMPLICATION:INFECTION	336	316	393	0.804	0.940	0.867	239	282	0.848	0.711	0.773	323	409	0.790	0.961	0.867	0.000	Rule Based	316	393	0.804	0.940	0.867
COMPLICATION:MEDICAL	171	142	213	0.667	0.830	0.740	132	175	0.754	0.772	0.763	155	250	0.620	0.906	0.736	-0.004	Rule Based	142	213	0.667	0.830	0.740
COMPLICATION:NEUROLOGICAL	28	24	58	0.414	0.857	0.558	17	29	0.586	0.607	0.596	25	68	0.368	0.893	0.521	-0.037	Rule Based	24	58	0.414	0.857	0.558
COMPLICATION:POSTOP BLEED	38	6	12	0.500	0.158	0.240	15	30	0.500	0.395	0.441	18	35	0.514	0.474	0.493	0.253	Combined	18	35	0.514	0.474	0.493
CRANIAL:ANEURYSM	116	89	295	0.302	0.767	0.433	48	114	0.421	0.414	0.417	100	376	0.266	0.862	0.407	-0.026	Rule Based	89	295	0.302	0.767	0.433
CRANIAL:ANEURYSM (UNRUPTURED)	238	101	131	0.771	0.424	0.547	179	312	0.574	0.752	0.651	199	383	0.520	0.836	0.641	0.094	Combined	199	383	0.520	0.836	0.641
CRANIAL:AVM	63	50	62	0.806	0.794	0.800	21	35	0.600	0.333	0.428	54	76	0.711	0.857	0.777	-0.023	Rule Based	50	62	0.806	0.794	0.800
CRANIAL:CAVERNOMA	16	14	21	0.667	0.875	0.757	4	8	0.500	0.250	0.333	14	23	0.609	0.875	0.718	-0.039	Rule Based	14	21	0.667	0.875	0.757
CRANIAL:CSF DISORDER	121	35	38	0.921	0.289	0.440	90	134	0.672	0.744	0.706	74	145	0.510	0.612	0.556	0.116	Combined	74	145	0.510	0.612	0.556
CRANIAL:CSF LEAK	38	26	65	0.400	0.684	0.505	12	17	0.706	0.316	0.437	32	72	0.444	0.842	0.581	0.076	Combined	32	72	0.444	0.842	0.581
CRANIAL:FISTULA	22	21	27	0.778	0.955	0.857	13	14	0.929	0.591	0.722	22	28	0.786	1.000	0.880	0.023	Rule Based	21	27	0.778	0.955	0.857
CRANIAL:NEOPLASIA	389	187	318	0.588	0.481	0.529	131	239	0.548	0.337	0.417	235	453	0.519	0.604	0.558	0.029	Rule Based	187	318	0.588	0.481	0.529
CRANIAL:NEOPLASIA:CYST	9	6	35	0.171	0.667	0.272	5	9	0.556	0.556	0.556	6	36	0.167	0.667	0.267	-0.005	Rule Based	6	35	0.171	0.667	0.272
CRANIAL:NEOPLASIA:GLIOMA	208	143	176	0.813	0.688	0.745	153	230	0.665	0.736	0.699	175	260	0.673	0.841	0.748	0.003	Rule Based	143	176	0.813	0.688	0.745
CRANIAL:NEOPLASIA:MENINGIOMA	192	157	188	0.835	0.818	0.826	144	181	0.796	0.750	0.772	173	223	0.776	0.901	0.834	0.008	Rule Based	157	188	0.835	0.818	0.826
CRANIAL:NEOPLASIA:METASTASIS	288	216	296	0.730	0.750	0.740	217	322	0.674	0.753	0.711	255	413	0.617	0.885	0.727	-0.013	Rule Based	216	296	0.730	0.750	0.740
CRANIAL:NEOPLASIA:PITUITARY	58	51	72	0.708	0.879	0.784	46	64	0.719	0.793	0.754	55	80	0.688	0.948	0.797	0.013	Rule Based	51	72	0.708	0.879	0.784
CRANIAL:NEOPLASIA:SCHWANNOMA	0	0	6	0.000	0.000	0.000	0	0	0.000	0.000	0.000	0	6	0.000	0.000	0.000	0.000	Rule Based	0	6	0.000	0.000	0.000
CRANIAL:NEOPLASIA:UNKNOWN	273	158	276	0.572	0.579	0.575	143	313	0.457	0.524	0.488	193	416	0.464	0.707	0.560	-0.015	Rule Based	158	276	0.572	0.579	0.575
CRANIAL:OTHER	86	18	523	0.034	0.209	0.058	32	54	0.593	0.372	0.457	37	557	0.066	0.430	0.114	0.056	Combined	37	557	0.066	0.430	0.114
CRANIAL:SAH (NON-ANEURYSMAL)	63	6	10	0.600	0.095	0.164	21	35	0.600	0.333	0.428	23	40	0.575	0.365	0.447	0.283	Combined	23	40	0.575	0.365	0.447
CRANIAL:SHUNT	57	47	103	0.456	0.825	0.587	39	64	0.609	0.684	0.644	52	114	0.456	0.912	0.608	0.021	Rule Based	47	103	0.456	0.825	0.587
CRANIAL:SKULL DEFECT	69	50	68	0.735	0.725	0.730	35	44	0.795	0.507	0.619	59	80	0.738	0.855	0.792	0.062	Combined	59	80	0.738	0.855	0.792
CRANIAL:TRAUMA	341	50	157	0.318	0.147	0.201	111	288	0.385	0.326	0.353	131	395	0.332	0.384	0.356	0.155	Combined	131	395	0.332	0.384	0.356
CRANIAL:TRAUMA:CONTUSIONS	510	421	533	0.790	0.825	0.807	373	685	0.545	0.731	0.624	455	843	0.540	0.892	0.673	-0.134	Rule Based	421	533	0.790	0.825	0.807
CRANIAL:TRAUMA:EDH	218	185	232	0.797	0.849	0.822	137	236	0.581	0.628	0.604	191	307	0.622	0.876	0.727	-0.095	Rule Based	185	232	0.797	0.849	0.822
CRANIAL:TRAUMA:ICH	151	111	232	0.478	0.735	0.579	74	288	0.257	0.490	0.337	125	438	0.285	0.828	0.424	-0.155	Rule Based	111	232	0.478	0.735	0.579
CRANIAL:TRAUMA:IVH	37	26	56	0.464	0.703	0.559	16	29	0.552	0.432	0.485	29	65	0.446	0.784	0.569	0.010	Rule Based	26	56	0.464	0.703	0.559
CRANIAL:TRAUMA:SAH	442	347	470	0.738	0.785	0.761	277	493	0.562	0.627	0.593	372	638	0.583	0.842	0.689	-0.072	Rule Based	347	470	0.738	0.785	0.761
CRANIAL:TRAUMA:SDH	1337	1210	1419	0.853	0.905	0.878	1109	1475	0.752	0.829	0.789	1270	1706	0.744	0.950	0.834	-0.044	Rule Based	1210	1419	0.853	0.905	0.878
CRANIAL:TRAUMA:SKULL FRACTURE	369	322	443	0.727	0.873	0.793	327	751	0.435	0.886	0.584	352	838	0.420	0.954	0.583	-0.210	Rule Based	322	443	0.727	0.873	0.793
CRANIAL:TRAUMA:TBI	757	471	532	0.885	0.622	0.731	386	497	0.777	0.510	0.616	445	742	0.600	0.588	0.594	-0.137	Rule Based	471	532	0.885	0.622	0.731
CRANIAL:UNCLASSIFIED	0	0	0	0.000	0.000	0.000	0	116	0.000	0.000	0.000	0	116	0.000	0.000	0.000	0.000	Rule Based	0	0	0.000	0.000	0.000
CRANIAL:VASCULAR	88	0	0	0.000	0.000	0.000	16	64	0.250	0.182	0.211	16	64	0.250	0.182	0.211	0.211	Combined	16	64	0.250	0.182	0.211
CRANIAL:VASCULAR:ICH	482	217	266	0.816	0.450	0.580	224	388	0.577	0.465	0.515	307	495	0.620	0.637	0.628	0.048	Combined	307	495	0.620	0.637	0.628
CRANIAL:VASCULAR:OCCLUSION	56	32	115	0.278	0.571	0.374	33	91	0.363	0.589	0.449	48	161	0.298	0.857	0.442	0.068	Combined	48	161	0.298	0.857	0.442
CRANIAL:VASCULAR:SAH	283	204	315	0.648	0.721	0.683	176	277	0.635	0.622	0.628	191	445	0.429	0.675	0.525	-0.158	Rule Based	204	315	0.648	0.721	0.683
EPILEPSY	34	26	68	0.382	0.765	0.510	23	57	0.404	0.676	0.506	28	76	0.368	0.824	0.509	-0.001	Rule Based	26	68	0.382	0.765	0.510
FUNCTIONAL DISORDER	3	0	0	0.000	0.000	0.000	1	3	0.333	0.333	0.333	1	3	0.333	0.333	0.333	0.333	Rule Based	0	0	0.000	0.000	0.000
FUNCTIONAL DISORDER:DYSTONIA	2	2	2	1.000	1.000	1.000	2	2	1.000	1.000	1.000	2	2	1.000	1.000	1.000	0.000	Rule Based	2	2	1.000	1.000	1.000
FUNCTIONAL DISORDER:PARKINSONS	4	4	6	0.667	1.000	0.800	4	4	1.000	1.000	1.000	4	6	0.667	1.000	0.800	0.000	Rule Based	4	6	0.667	1.000	0.800
FUNCTIONAL DISORDER:SPASTICITY	1	1	2	0.500	1.000	0.667	0	0	0.000	0.000	0.000	1	2	0.500	1.000	0.667	0.000	Rule Based	1	2	0.500	1.000	0.667
FUNCTIONAL DISORDER:TREMOR	2	2	2	1.000	1.000	1.000	1	1	1.000	0.500	0.667	2	2	1.000	1.000	1.000	0.000	Rule Based	2	2	1.000	1.000	1.000
HYDROCEPHALUS	50	30	83	0.361	0.600	0.451	22	41	0.537	0.440	0.484	33	98	0.337	0.660	0.446	-0.005	Rule Based	30	83	0.361	0.600	0.451
OTHER	0	0	0	0.000	0.000	0.000	0	54	0.000	0.000	0.000	0	54	0.000	0.000	0.000	0.000	Rule Based	0	0	0.000	0.000	0.000
PAIN	33	30	55	0.545	0.909	0.681	12	16	0.750	0.364	0.490	30	57	0.526	0.909	0.666	-0.015	Rule Based	30	55	0.545	0.909	0.681
PERIPHERAL NERVE	25	11	25	0.440	0.440	0.440	11	24	0.458	0.440	0.449	14	40	0.350	0.560	0.431	-0.009	Rule Based	11	25	0.440	0.440	0.440
PERIPHERAL NERVE:NEOPLASIA	7	4	5	0.800	0.571	0.666	1	1	1.000	0.143	0.250	5	6	0.833	0.714	0.769	0.103	Rule Based	4	5	0.800	0.571	0.666
PERIPHERAL NERVE:TRAUMA	8	0	0	0.000	0.000	0.000	0	0	0.000	0.000	0.000	0	0	0.000	0.000	0.000	0.000	Rule Based	0	0	0.000	0.000	0.000
SPINE:AVM	3	2	3	0.667	0.667	0.667	0	0	0.000	0.000	0.000	2	3	0.667	0.667	0.667	0.000	Rule Based	2	3	0.667	0.667	0.667
SPINE:CANAL STENOSIS	210	105	149	0.705	0.500	0.585	140	277	0.505	0.667	0.575	150	302	0.497	0.714	0.586	0.001	Rule Based	105	149	0.705	0.500	0.585
SPINE:CAVERNOMA	2	2	2	1.000	1.000	1.000	0	0	0.000	0.000	0.000	2	2	1.000	1.000	1.000	0.000	Rule Based	2	2	1.000	1.000	1.000
SPINE:CSF DISORDER	17	9	21	0.429	0.529	0.474	9	15	0.600	0.529	0.562	13	27	0.481	0.765	0.591	0.117	Combined	13	27	0.481	0.765	0.591
SPINE:CSF LEAK	17	3	4	0.750	0.176	0.285	13	56	0.232	0.765	0.356	13	57	0.228	0.765	0.351	0.066	Combined	13	57	0.228	0.765	0.351
SPINE:DAVF	4	3	4	0.750	0.750	0.750	3	3	1.000	0.750	0.857	3	4	0.750	0.750	0.750	0.000	Rule Based	3	4	0.750	0.750	0.750
SPINE:DEGENERATIVE	820	444	512	0.867	0.541	0.666	627	762	0.823	0.765	0.793	672	872	0.771	0.820	0.795	0.129	Combined	672	872	0.771	0.820	0.795
SPINE:HAEMATOMA	8	4	7	0.571	0.500	0.533	3	5	0.600	0.375	0.462	6	10	0.600	0.750	0.667	0.134	Rule Based	4	7	0.571	0.500	0.533
SPINE:NEOPLASIA	211	115	132	0.871	0.545	0.670	121	192	0.630	0.573	0.600	157	242	0.649	0.744	0.693	0.023	Rule Based	115	132	0.871	0.545	0.670
SPINE:OTHER	183	128	783	0.163	0.699	0.264	73	146	0.500	0.399	0.444	149	845	0.176	0.814	0.289	0.025	Rule Based	128	783	0.163	0.699	0.264
SPINE:SYRINX	14	11	16	0.688	0.786	0.734	11	13	0.846	0.786	0.815	13	18	0.722	0.929	0.813	0.079	Combined	13	18	0.722	0.929	0.813
SPINE:TRAUMA	179	83	100	0.830	0.464	0.595	131	301	0.435	0.732	0.546	92	344	0.267	0.514	0.351	-0.244	Rule Based	83	100	0.830	0.464	0.595
SPINE:TRAUMA:DISCO-LIGAMENTOUS	82	39	94	0.415	0.476	0.443	53	88	0.602	0.646	0.623	57	154	0.370	0.695	0.483	0.040	Combined	57	154	0.370	0.695	0.483
SPINE:TRAUMA:FRACTURE	1531	1368	1480	0.924	0.894	0.909	1101	1148	0.959	0.719	0.822	1444	1600	0.903	0.943	0.923	0.014	Rule Based	1368	1480	0.924	0.894	0.909
SPINE:TRAUMA:OTHER	11	0	0	0.000	0.000	0.000	6	6	1.000	0.545	0.706	6	6	1.000	0.545	0.706	0.706	Combined	6	6	1.000	0.545	0.706
SPINE:VASCULAR	15	10	12	0.833	0.667	0.741	3	3	1.000	0.200	0.333	11	13	0.846	0.733	0.785	0.044	Combined	11	13	0.846	0.733	0.785
SYRINX	3	0	0	0.000	0.000	0.000	2	2	1.000	0.667	0.800	2	2	1.000	0.667	0.800	0.800	Rule Based	0	0	0.000	0.000	0.000
ULNAR NERVE	29	25	27	0.926	0.862	0.893	15	16	0.938	0.517	0.667	28	30	0.933	0.966	0.949	0.056	Combined	28	30	0.933	0.966	0.949
Total and Averages	**12023**	**8159**	**12023**	0.585	0.600	0.562	**7631**	**12023**	0.588	0.515	0.524	**9471**	**17176**	0.541	0.727	0.597	**2.514**		**8913**	**13685**	0.583	0.671	0.601
Micro Averages				**0.679**	**0.679**	**0.679**			**0.635**	**0.635**	**0.635**			**0.551**	**0.788**	**0.649**					**0.651**	**0.741**	**0.693**
Macro Weighted Averages				**0.777**	**0.754**	**0.749**			**0.695**	**0.680**	**0.672**			**0.643**	**0.827**	**0.709**					**0.746**	**0.780**	**0.751**

Appendix B

SVM-Based predictions for rule-based other predictions

DiagnosisAuditCode	ClassTotal	Matched	Predicted	Precision	Recall	FScore
CARPAL TUNNEL	2	2	2	1.000	1.000	1.000
COMPLICATION	94	56	57	0.982	0.596	0.742
COMPLICATION:DEVICE	2	2	4	0.500	1.000	0.667
COMPLICATION:INFECTION	12	8	10	0.800	0.667	0.727
COMPLICATION:MEDICAL	18	15	17	0.882	0.833	0.857
COMPLICATION:NEUROLOGICAL	2	1	1	1.000	0.500	0.667
COMPLICATION:POSTOP BLEED	7	2	3	0.667	0.286	0.400
CRANIAL:ANEURYSM	0	0	2	0.000	0.000	0.000
CRANIAL:CSF DISORDER	0	0	1	0.000	0.000	0.000
CRANIAL:CSF LEAK	0	0	1	0.000	0.000	0.000
CRANIAL:NEOPLASIA:GLIOMA	0	0	1	0.000	0.000	0.000
CRANIAL:OTHER	0	0	3	0.000	0.000	0.000
CRANIAL:SKULL DEFECT	0	0	1	0.000	0.000	0.000
CRANIAL:TRAUMA	0	0	1	0.000	0.000	0.000
CRANIAL:TRAUMA:CONTUSIONS	0	0	3	0.000	0.000	0.000
CRANIAL:TRAUMA:ICH	0	0	2	0.000	0.000	0.000
CRANIAL:TRAUMA:SDH	1	0	7	0.000	0.000	0.000
CRANIAL:TRAUMA:SKULL FRACTURE	5	5	8	0.625	1.000	0.769
CRANIAL:TRAUMA:TBI	0	0	3	0.000	0.000	0.000
CRANIAL:UNCLASSIFIED	0	0	1	0.000	0.000	0.000
CRANIAL:VASCULAR	0	0	1	0.000	0.000	0.000
CRANIAL:VASCULAR:ICH	0	0	7	0.000	0.000	0.000
EPILEPSY	3	3	3	1.000	1.000	1.000
FUNCTIONAL DISORDER	1	1	1	1.000	1.000	1.000
HYDROCEPHALUS	3	3	7	0.429	1.000	0.600
PAIN	5	2	2	1.000	0.400	0.571
PERIPHERAL NERVE	2	2	2	1.000	1.000	1.000
SPINE:CSF DISORDER	0	0	3	0.000	0.000	0.000
SPINE:OTHER	0	0	1	0.000	0.000	0.000
SPINE:TRAUMA	0	0	1	0.000	0.000	0.000
SPINE:TRAUMA:DISCO-LIGAMENTOUS	0	0	2	0.000	0.000	0.000
SPINE:TRAUMA:FRACTURE	0	0	1	0.000	0.000	0.000
ULNAR NERVE	2	0	0	0.000	0.000	0.000
Total and Averages	**159**	**102**	**159**	0.287	0.280	0.266
Micro Averages				**0.642**	**0.642**	**0.642**
Macro Weighted Averages				**0.906**	**0.697**	**0.763**

Appendix C

Summary Scores Comparisons

Summary scores when excluding Rule-Based OTHER predictions

		Rule Based			SVM Based			Combined			Hybrid		
	Class	Matched	Predicted		Matched	Predicted		Matched	Predicted		Matched	Predicted	
Totals	12023	8159	12023		7631	12023		9471	17176		8913	13685	
		Precision	Recall	F Score	Precision	Recall	F Score	Precision	Recall	F Score	Precision	Recall	F Score
Micro averages		0.679	0.679	0.679	0.635	0.635	0.635	0.551	0.788	0.649	0.651	0.741	0.693
Macro Weighted averages		0.777	0.754	0.749	0.695	0.680	0.672	0.643	0.827	0.709	0.746	0.780	0.751

Summary scores when including Rule-Based OTHER predictions

		Rule Based			SVM Based			Combined			Hybrid		
	Total	Matched	Predicted		Matched	Predicted		Matched	Predicted		Matched	Predicted	
Totals	12182	8159	12182		7731	12182		9571	17494		9007	14006	
		Precision	Recall	F Score	Precision	Recall	F Score	Precision	Recall	F Score	Precision	Recall	F Score
Micro averages		0.670	0.670	0.670	0.635	0.635	0.635	0.547	0.786	0.645	0.643	0.739	0.688
Macro Weighted averages		0.777	0.749	0.746	0.694	0.679	0.671	0.644	0.824	0.708	0.745	0.778	0.749

Differences when including Rule-Based OTHER predictions

		Rule Based			SVM Based			Combined			Hybrid		
	Total	Matched	Predicted		Matched	Predicted		Matched	Predicted		Matched	Predicted	
	159	0	159		100	159		100	318		94	321	
		Precision	Recall	F Score	Precision	Recall	F Score	Precision	Recall	F Score	Precision	Recall	F Score
Micro averages		-0.009	-0.009	-0.009	0.000	0.000	0.000	-0.004	-0.002	-0.004	-0.008	-0.002	-0.005
Macro Weighted averages		0.000	-0.005	-0.003	-0.001	-0.001	-0.001	0.001	-0.003	-0.001	-0.001	-0.002	-0.002

Appendix D

Distribution of OTHER

Prediction	Rule-Based		SVM-Based		In Common	
	Count	Percent	Count	Percent	Count	Percent
OTHER	134	22.26%	186	30.90%	67	11.13%
COMPLICATION:INFECTION	59	9.80%	28	4.65%	27	4.49%
COMPLICATION	47	7.81%	35	5.81%	13	2.16%
COMPLICATION:MEDICAL	35	5.81%	29	4.82%	14	2.33%
CRANIAL:NEOPLASIA	25	4.15%	6	1.00%	5	0.83%
PAIN	24	3.99%	11	1.83%	9	1.50%
CRANIAL:TRAUMA:SDH	24	3.99%	36	5.98%	16	2.66%
CRANIAL:SHUNT	22	3.65%	7	1.16%	6	1.00%
CRANIAL:VASCULAR:OCCLUSION	19	3.16%	9	1.50%	7	1.16%
SPINE:OTHER	18	2.99%	9	1.50%	1	0.17%
CRANIAL:OTHER	17	2.82%	13	2.16%	1	0.17%
CRANIAL:VASCULAR:SAH	16	2.66%	7	1.16%	3	0.50%
SPINE:DEGENERATIVE	14	2.33%	7	1.16%	3	0.50%
CRANIAL:TRAUMA:SKULL FRACTURE	10	1.66%	22	3.65%	8	1.33%
CRANIAL:TRAUMA:TBI	10	1.66%	16	2.66%	0	0.00%
EPILEPSY	10	1.66%	7	1.16%	6	1.00%
COMPLICATION:NEUROLOGICAL	10	1.66%	3	0.50%	2	0.33%
CRANIAL:TRAUMA:SAH	10	1.66%	15	2.49%	4	0.66%
SPINE:TRAUMA:FRACTURE	8	1.33%	5	0.83%	1	0.17%
CRANIAL:NEOPLASIA:CYST	7	1.16%	0	0.00%	0	0.00%
CRANIAL:VASCULAR:ICH	7	1.16%	9	1.50%	3	0.50%
CRANIAL:SKULL DEFECT	7	1.16%	6	1.00%	2	0.33%
CRANIAL:NEOPLASIA:METASTASIS	7	1.16%	12	1.99%	4	0.66%
CRANIAL:ANEURYSM	7	1.16%	1	0.17%	1	0.17%
CRANIAL:NEOPLASIA:UNKNOWN	7	1.16%	9	1.50%	3	0.50%
CHIARI	6	1.00%	0	0.00%	0	0.00%
CRANIAL:TRAUMA:EDH	4	0.66%	3	0.50%	1	0.17%
SPINE:TRAUMA	4	0.66%	1	0.17%	0	0.00%
FUNCTIONAL DISORDER:SPASTICITY	4	0.66%	0	0.00%	0	0.00%
HYDROCEPHALUS	4	0.66%	4	0.66%	2	0.33%
CRANIAL:CSF LEAK	3	0.50%	0	0.00%	0	0.00%
CRANIAL:TRAUMA:CONTUSIONS	3	0.50%	10	1.66%	1	0.17%
PERIPHERAL NERVE	3	0.50%	1	0.17%	1	0.17%
CARPAL TUNNEL	2	0.33%	2	0.33%	2	0.33%
CRANIAL:NEOPLASIA:MENINGIOMA	2	0.33%	0	0.00%	0	0.00%
CRANIAL:NEOPLASIA:PITUITARY	2	0.33%	0	0.00%	0	0.00%
FUNCTIONAL DISORDER:PARKINSONS	2	0.33%	1	0.17%	1	0.17%
SPINE:VASCULAR	2	0.33%	0	0.00%	0	0.00%
CRANIAL:ANEURYSM (UNRUPTURED)	1	0.17%	4	0.66%	0	0.00%
CRANIAL:CSF DISORDER	1	0.17%	16	2.66%	0	0.00%
ULNAR NERVE	1	0.17%	2	0.33%	0	0.00%
CRANIAL:NEOPLASIA:GLIOMA	1	0.17%	2	0.33%	1	0.17%
COMPLICATION:DEVICE	1	0.17%	4	0.66%	1	0.17%
SPINE:CSF DISORDER	1	0.17%	2	0.33%	0	0.00%
SPINE:CANAL STENOSIS	1	0.17%	1	0.17%	0	0.00%
SPINE:CSF LEAK	0	0.00%	2	0.33%	0	0.00%
COMPLICATION:POSTOP BLEED	0	0.00%	1	0.17%	0	0.00%
SPINE:TRAUMA:DISCO-LIGAMENTOUS	0	0.00%	4	0.66%	0	0.00%
SPINE:NEOPLASIA	0	0.00%	5	0.83%	0	0.00%
CRANIAL:TRAUMA:ICH	0	0.00%	6	1.00%	0	0.00%
CRANIAL:TRAUMA	0	0.00%	12	1.99%	0	0.00%
CRANIAL:UNCLASSIFIED	0	0.00%	21	3.49%	0	0.00%
CRANIAL:VASCULAR	0	0.00%	9	1.50%	0	0.00%
FUNCTIONAL DISORDER	0	0.00%	1	0.17%	0	0.00%
Grand Total	**602**	**100.00%**	**602**	**100.00%**	**216**	**35.88%**

Appendix E

Distribution of CRANIAL:UNCLASSIFIED

Prediction	Rule-Based		SVM-Based		In Common	
	Count	Percent	Count	Percent	Count	Percent
CRANIAL:TRAUMA:SDH	47	18.43%	43	16.86%	36	14.12%
CRANIAL:OTHER	23	9.02%	10	3.92%	0	0.00%
CRANIAL:UNCLASSIFIED	22	8.63%	58	22.75%	9	3.53%
CRANIAL:VASCULAR:ICH	15	5.88%	14	5.49%	6	2.35%
CRANIAL:NEOPLASIA:UNKNOWN	12	4.71%	12	4.71%	9	3.53%
CRANIAL:TRAUMA:CONTUSIONS	11	4.31%	10	3.92%	5	1.96%
CRANIAL:SKULL DEFECT	9	3.53%	13	5.10%	8	3.14%
CRANIAL:TRAUMA:SKULL FRACTURE	8	3.14%	8	3.14%	4	1.57%
CRANIAL:TRAUMA:SAH	8	3.14%	12	4.71%	3	1.18%
CRANIAL:VASCULAR:SAH	8	3.14%	3	1.18%	2	0.78%
CRANIAL:SHUNT	8	3.14%	2	0.78%	2	0.78%
CRANIAL:NEOPLASIA:METASTASIS	7	2.75%	4	1.57%	3	1.18%
CRANIAL:VASCULAR:OCCLUSION	7	2.75%	3	1.18%	2	0.78%
CRANIAL:NEOPLASIA	7	2.75%	5	1.96%	2	0.78%
COMPLICATION:INFECTION	7	2.75%	2	0.78%	2	0.78%
CRANIAL:TRAUMA:TBI	6	2.35%	2	0.78%	1	0.39%
EPILEPSY	5	1.96%	5	1.96%	4	1.57%
CRANIAL:NEOPLASIA:CYST	5	1.96%	0	0.00%	0	0.00%
CRANIAL:CSF LEAK	4	1.57%	0	0.00%	0	0.00%
CRANIAL:NEOPLASIA:GLIOMA	4	1.57%	2	0.78%	2	0.78%
CHIARI	4	1.57%	0	0.00%	0	0.00%
CRANIAL:TRAUMA:EDH	3	1.18%	6	2.35%	2	0.78%
CRANIAL:TRAUMA:IVH	3	1.18%	0	0.00%	0	0.00%
CRANIAL:ANEURYSM	2	0.78%	2	0.78%	0	0.00%
CRANIAL:NEOPLASIA:PITUITARY	2	0.78%	1	0.39%	1	0.39%
CRANIAL:ANEURYSM (UNRUPTURED)	2	0.78%	1	0.39%	0	0.00%
SPINE:OTHER	2	0.78%	0	0.00%	0	0.00%
PAIN	2	0.78%	1	0.39%	1	0.39%
CRANIAL:AVM	2	0.78%	2	0.78%	1	0.39%
CRANIAL:TRAUMA	2	0.78%	7	2.75%	1	0.39%
HYDROCEPHALUS	1	0.39%	1	0.39%	1	0.39%
COMPLICATION	1	0.39%	2	0.78%	0	0.00%
COMPLICATION:NEUROLOGICAL	1	0.39%	1	0.39%	0	0.00%
CRANIAL:NEOPLASIA:MENINGIOMA	1	0.39%	0	0.00%	0	0.00%
SPINE:VASCULAR	1	0.39%	0	0.00%	0	0.00%
CRANIAL:CSF DISORDER	1	0.39%	7	2.75%	0	0.00%
CRANIAL:CAVERNOMA	1	0.39%	0	0.00%	0	0.00%
CRANIAL:TRAUMA:ICH	1	0.39%	9	3.53%	0	0.00%
SPINE:CSF LEAK	0	0.00%	2	0.78%	0	0.00%
CRANIAL:VASCULAR	0	0.00%	3	1.18%	0	0.00%
SPINE:NEOPLASIA	0	0.00%	1	0.39%	0	0.00%
COMPLICATION:DEVICE	0	0.00%	1	0.39%	0	0.00%
Grand Total	**255**	**100.00%**	**255**	**100.00%**	**107**	**41.96%**

Appendix F

Micro and Macro Averaging Explanation

To illustrate the difference between micro and macro averaging methods, consider the following small example, the classes sum to 45, there is 1 prediction per record so therefore also 45 predictions, but only 26 matches. Total Precision is Total Matched/Total Predicted = 26/45 = 0.578; Total Recall = Total Matched/Class Total = 26/45 = 0.578; Total F-Score = 1/((0.5/Total Precision) + (0.5/Total Recall)) = 1/((0.5/0.578)+(0.5/0.578)) = 0.578. However, if the individual calculations for Precision, Recall and F-Score for each class (using the same formulae as above but based on each class total) are extended by the number of matches per class, then those figures are summed and finally divided by the total matched, a weighted average is obtained, and is more reflective of the general accuracy of the predictions – Precision is 0.626, Recall is 0.587, and F-Score is 0.598.

	Class Total	Matched	Predicted	Precision	Precision Ext	Recall	Recall Ext	F-Score	F-Score Ext
	15	10	12	0.833	8.333	0.667	6.667	0.741	7.407
	10	6	15	0.400	2.400	0.600	3.600	0.480	2.880
	20	10	18	0.556	5.556	0.500	5.000	0.526	5.263
Totals	45	26	45		16.289		15.267		15.551
Micro Averages				0.578		0.578		0.578	
Macro Weighted Averages					0.626		0.587		0.598