

University of Southern Queensland
Faculty of Health, Engineering and Sciences

Natural Language Processing to support Nurse-to-Patient allocation in acute care

A dissertation submitted by

Daniel Price

in fulfilment of the requirements of

ENG4111 and 4112 Research Project

towards the degree of

Bachelor of Engineering (Honours) (Computer Systems)

Submitted October 2021

Abstract

The modern Australian health care system demands a fast and high interpersonal level of care for all patients, with a stronger focus than ever on patient outcomes. A significant level of responsibility for these improved outcomes and reduced hospital stays is placed heavily on Nurses and the nursing profession. While nursing staff are supported by improved technology and workflow procedures it can be seen that some of these systems require heavy admin work and/or non-clinical skills to be effective, reducing time spent with patients and therefore potentially prolonging hospital stays.

Research into optimal staff management has provided evidence that effective distribution of staff skills can improve patient outcomes, staff satisfaction, and reduce costs. Healthcare workers appreciate a consistent and fair schedule that they can rely on. While systems and computer-based programs exist for staff-to-patient allocation the process of assigning a staff member to a patient based on skills and patient needs is a time-consuming process.

This project aims to provide a solution to the implementation of the optimal model of care through the use of Natural Language Processing (NLP) and Machine Learning (ML). By automating the staff-to-patient allocation process the optimal model of care can be implemented with little to no admin overheads resulting in nursing staff spending more time with patients and therefore improved patient outcomes.

NLP algorithms and techniques including NER and TF-IDF for Topic modelling have been explored and analysed to determine if accurate extraction of critical patient information from nurse progress notes can be achieved. A Machine Learning neural network based on python's TensorFlow and Keras ML libraries were also explored. The SpaCy NER model designed in the project return an accuracy of 89% for competency extraction and has proven to be the most reliable. This showed that an automation of staff-to-patient allocation is plausible provided a ward specific allocation is adopted.

University of Southern Queensland
Faculty of Health, Engineering and Sciences

ENG4111 & ENG4112 Research Project

Limitations of Use

The Council of the University of Southern Queensland, its Faculty of Health, Engineering and Sciences, and the staff of the University of Southern Queensland, do not accept any responsibility for the truth, accuracy or completeness of material contained within or associated with this dissertation.

Persons using all or any part of this material do so at their own risk, and not at the risk of the Council of the University of Southern Queensland, its Faculty of Health, Engineering and Sciences or the staff of the University of Southern Queensland.


This dissertation reports an educational exercise and has no purpose or validity beyond this exercise. The sole purpose of the course pair entitled “Research Project” is to contribute to the overall education within the student’s chosen degree program. This document, the associated hardware, software, drawings, and any other material set out in the associated appendices should not be used for any other purpose: if they are so used, it is entirely at the risk of the user.

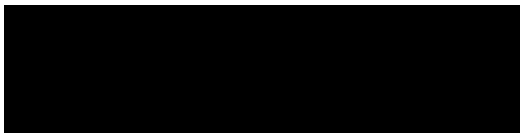
Certification

I certify that the ideas, designs, and experimental work, results, analyses and conclusions set out in this dissertation are entirely my own effort, except where otherwise indicated and acknowledged.

I further certify that the work is original and has not been previously submitted for assessment in any other course or institution, except where specifically stated.

Daniel Price

Student 



Signature

21/20/2021

Date

Acknowledgements

I would like to thank all staff involved from the Research and Ethics department from the Gold Coast University Hospital for their support and knowledge. Thanks, must also go to Rauland Australia for supporting my studies and my supervisor Mark Phythian.

I would also like to thank my fiancé for her support and knowledge as a Clinical Nurse throughout the project and during the completion of my studies.

Table of Contents

List of Tables.....	8
List of Figures	8
Acronyms	9
Chapter 1. Introduction.....	10
1.1 Project Aim.....	11
1.2 Project Objectives	11
1.3 Overview of the Dissertation.....	11
Chapter 2. Literature Review.....	12
2.1 Chapter Overview	12
2.1 Introduction	12
2.2 Nursing	12
2.3 Nurse-to-patient ratios.....	13
2.4 Staffing and Rosters.....	13
2.5 Patient-to-staff allocation process.....	14
2.6 Patient-to-staff allocation risks	15
2.7 Patient risks and outcomes	15
2.8 Chapter Summary	16
Chapter 3. Natural Language Processing.....	17
3.1 Chapter Overview	17
3.2 Python programming language	17
3.3 Introduction to Natural Language Processing (NLP).....	17
3.4 Data Cleaning	17
3.5 Punctuation	18
3.6 Tokenization.....	18
3.7 Stop Words Removal	18
3.8 Lemmatization	18
3.9 Part of Speech Tagging.....	19
3.10 Term Frequency-Inverse Document Frequency (TF-IDF).....	19
3.11 Non-negative Matrix Factorization	20
3.12 Tesnorflow	21
3.13 Keras.....	21
3.14 Multiclass text classification with Tensflow and Keras	22
3.15 LSTM – Long Short-Term Memory.....	22
3.16 SpaCy	23

3.17	SpaCy – Named Entity Recognition (NER)	23
3.18	Chapter Summary	23
Chapter 4.	Data Requirements and Manipulation	24
4.1	Chapter Overview	24
4.2	Data Type	24
4.2.1	Nurse Progress Note	24
4.2.2	Competency key word(s)	25
4.2.3	Staff Experience and Competencies	26
4.3	Data Collection	28
4.3.2	Developing hypothetical data for multiclass text classification	28
4.4	Progress Note Data	28
4.5	Chapter Summary	30
Chapter 5.	System Layout and Testing Scenarios	31
5.1	Chapter Overview	31
5.2	General System Layout	31
5.3	Staff function	31
5.3	Testing Scenarios	32
5.3.1	Model 1 – Topic Modelling tf-idf	32
5.3.2	Model 2 – Topic Modelling with tf-idf and NMF	33
5.3.3	Model 3 – Multiclass text classification with Keras and TesnorFlow	33
5.3.4	Model 4 – Named Entity Recognition with Spacy	34
5.4	Chapter Summary	35
Chapter 6.	Evaluation and Testing	36
6.1	Chapter Overview	36
6.2	Model 1 – Topic Modelling with TF-IDF	36
6.3	Model 2 – Topic Modelling with TF-IDF and NMF	40
6.4	Model 3 – Multiclass text classification with Keras and TesnorFlow	43
6.5	Model 4 – Named Entity Recognition with Spacy	46
6.6	Model Accuracy	48
6.7	Chapter Summary	49
Chapter 7.	Conclusions and Further Work	50
7.1	Achievement of Project Objectives:	50
7.2	Further Work	51
References	52
Appendix A	55
Project specifications	55

Appendix B.....	57
Example Project Notes	57
Appendix C.....	64
Staff Competency Files	64
Appendix D	68
Key Words List	68
Appendix E.....	72
Appendix F.....	78
Appendix G	79
Python Scripts.....	79

List of Tables

Table 1: Staff competency table.....	27
Table 2: Progress notes required competencies	29
Table 3: TF-IDF results	48
Table 4: TF-IDF+NMF results	49
Table 5: SpaCy NER results	49

List of Figures

Figure 1:Nurse progress note	25
Figure 2:Spacy model layout	34
Figure 3:NMF topic modelling results	41
Figure 4:Multiclass model summary	43
Figure 5:Multiclass Model outcomes	44
Figure 6:Multiclass model prediction results	45

Acronyms

NLP	Natural Language Processing
ML	Machine Learning
NER	Named Entity Recognition
TF-IDF	Term Frequency-Inverse Document Frequency
NMF	Non-Negative Matrix Factorisation
LSTM	Long Short-Term Memory
RN	Registered Nurse
CN	Clinical Nurse
NUM	Nurse Unit Manager
EN	Enrolled Nurse
AIN	Assistant in Nursing
AI	Artificial Intelligence
CSV	Comma Separated Values
SVM	Support Vector Machines
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
ECG	Electrocardiogram
GCUH	Gold Coast University Hospital
CCU	Coronary Care Unit
STEMI	ST-Elevation Myocardial Infarction

Chapter 1. Introduction

Traditional methods of nurse staff assignments continue to be used to this day and suggest that a volumetric approach based on a census at a single point in time, midnight, can accurately outline patient needs and their associated level of care for the following shift and even the next day. This approach to staffing assignments may have been successful in past years but the modern health care system expects a significantly higher level of care in shorter periods.

Research into optimal staff management has provided evidence that effective distribution of staff skills can improve patient outcomes, staff satisfaction, and reduce costs. Healthcare workers appreciate a consistent and fair schedule that they can rely on. But it is often elusive, and that impacts everything from the personal lives of clinicians to patient satisfaction and outcomes (AMN HealthCare 2020). Research suggests that undoubtedly there is a benefit to switching to an optimal staff model, however, it remains unclear as to why this has not seen a more widespread adoption, there appears to be limited research in the optimal model adoption, however, personal industry experience (5+ years) provides one theory that the technology has not been available to support the requirements of this model. The administration of such in depth modelling and staff assignments is onerous and is quickly cast aside for a more time effective method. Current staff assignment programs exist but must be administered manually, patient acuity must first be assessed and reviewed with relation to the staff on shift. Once a patient-to-staff allocation has been completed this must be manually entered into an allocation program prior to the shift beginning.

The inability to predict the number of patients leaving and arriving at any one time causes the allocation process to be time consuming and stressful for staff and even patients. When entering a hospital, nursing staff are often the first to engage and initiate a patient's recovery journey, ensuring that the patient is comfortable, begin the admission process and provide an initial diagnosis. Regardless of the circumstances surrounding a patient's arrival and their condition, an accident arriving in the emergency department, afterhours or an elective surgery in business hours, a nurse must admit the patient and ensure they have the correct care and support for their stay. Depending on the severity of the patient's condition, care can range from simple duties such as delivering food and managing medication to medically complex duties, such as controlling medication and cardiopulmonary resuscitation (Milstein & Schreyoegg 2020).

Research into the use of NLP in medicine and medical applications suggests a positive outcome can be achieved if implemented in a subtle but effective manner. The medical vocabulary is vast and ever growing forcing the need for machine support within the industry. The ability for machines to understand natural language and quickly provide feedback or extract critical information from large datasets could significantly reduce time spent on admin related work and give time back to clinicians for patient care.

The motivation for this project was developed over many experiences and interactions with nursing staff and their frustration with the loss of time to admin related work, time better spent caring for patients. Correct staff assignment can undoubtedly provide improved patient outcomes and if implementation can be achieved through an automated solution it stands to reason that improved patient outcomes will follow.

1.1 Project Aim

To design and implement a Machine Learning model using Natural Language Processing algorithms for the purpose of autonomously completing the staff-to-patient allocation process in an acute care facility.

1.2 Project Objectives

1. Research and define the results relating to patient outcomes and staff satisfaction when implementing optimal staff-to-patient allocation.
2. Research and test the techniques of Natural Language Processing and Machine Learning models in Health Care.
3. Design and implement an automate solution to the staff-to-patient allocation process using techniques and models outlined in objective 2.
4. Analyse the results of the program for accuracy, scalability and change to existing workflow to determine plausibility for real world development.

1.3 Overview of the Dissertation

The following provides an overview of the dissertation and its chapters.

Chapter 2 presents the literature review giving a brief outline of the nursing profession and how improved patient outcomes can be achieved based on the adoption of the optimal model of care.

Chapter 3 introduces the concepts of Natural Language Processing and Machine Learning in Python.

Chapter 4 provides an overview of the data required, its format and how it was obtained.

Chapter 5 outlines each of techniques tested, the testing scenarios and model parameters

Chapter 6 presents and discusses the results of the techniques tested and evaluates accuracy of each model

Chapter 7 presents the project conclusion and outlines areas for improvement and further work.

Chapter 2. Literature Review

2.1 Chapter Overview

Chapter 2 provides a literature review into the nursing practice, acute care staffing requirements and appropriate staffing skill mix in Australia. It also investigates the success and risks of varying staff assignment models and the trade-offs for an increased level of detail when completing the staff-to-patient allocation. Finally it also considers the patient outcomes based on the staffing model chosen.

2.1 Introduction

The World Health Organization alerts that managing and deploying personnel resources efficiently are key challenges for the healthcare industry in the coming decennia due to the increasing demand for care and healthcare providers (Maenhout & Vanhoucke 2013). Providing safe, high quality health services is an expectation for all service providers (Queensland Health 2016). In order to ensure excellent health care is provided, the appropriate skill mix of nurses on a shift is critical and should be distributed in an effective manner (Queensland Health 2016).

Research has shown that inappropriate skill mix, poor and unattractive schedules, poor practice environments and high workloads are identified as important factors leading to decreased patient safety, staff discontentment and a high nursing turnover (Maenhout & Vanhoucke 2013). Current techniques for allocating appropriate nurses to wards and patients can be optimised with the use of new technology. This literature review will discuss current patient nurse allocation systems implemented with Australian hospitals as well as explore the drawbacks of current systems and why there is a need for change. The literature review will detail bed management technology, technology in health, NLP techniques, NLP algorithms and neural networks. Limitations and considerations of the research will also be highlighted.

2.2 Nursing

A registered nurse (RN) has a complicated and demanding role within the health care system. Their work is complex and both intellectually and managerially stringent. The more complex, cognitive and organisational activities of nurses are often unobserved and unappreciated because they are often done concurrently with the task-oriented work of nurses (Flinter, Hsu, Cromp, Ladden & Wagner 2017). While being task focussed, RNs are also constantly assessing, monitoring and detecting deterioration in their patients. They are risk aware and monitor for falls, pressure injuries and other complications that can arise within their care (Flinter, Hsu, Cromp, Ladden & Wagner 2017).

Based on these assessments, nurses are expected to initiate appropriate nursing interventions or escalate concerns to the medical team for follow up and consultations (Flinter, Hsu, Cromp, Ladden & Wagner 2017). RNs monitor patients pain levels and take action to control it. As well as providing front line care, nurses also assist with secondary and tertiary prevention of diseases (Flinter, Hsu, Cromp, Ladden & Wagner 2017). Education of patients and family members is a vital role undertaken by RNs which assists in self-care and disease management after discharge. RNs also provide psychological support for patients dealing with serious illness (Flinter, Hsu, Cromp, Ladden & Wagner 2017). Nurses play a critical role within the multidisciplinary team. They backstop consultants and pharmacists by ensuring medications are ordered and prescribed correctly and are responsible for correct administration of medications (Flinter, Hsu, Cromp, Ladden & Wagner 2017). They often serve as the principal coordinators of care for their patients and are important patient advocates. They refer patients for dietician, physiotherapy, occupational therapy and

psychology consults when they feel appropriate (Flinter, Hsu, Crompt, Ladden & Wagner 2017). Along with patient care, experienced RNs also have a responsibility to educate junior nurses and nursing students and are often shadowed by a junior colleague throughout their shift. These activities are carried out for each patient by nurses responsible for one, two, four, six or more patients at the same time, a situation that imposes substantial managerial demands on frontline nurses (Flinter, Hsu, Crompt, Ladden & Wagner 2017).

2.3 Nurse-to-patient ratios

Health services need sufficient nurses to ensure that patient care can be provided safely and effectively. A nurse-to-patient ratio is the minimum number of nurses working on a particular ward, unit or department, in relation to the number of patients on the ward (Queensland Health 2016). The ratio is calculated on the basis of patient acuity: the greater the level of acuity, the higher the number of nurses required to provide safe care (Queensland Health 2016). Legislated minimum ratios came into effect in Australia in 2016 (Queensland Health 2016). The Queensland Government outlines a ratio of one nurse to four patients on a medical and surgical ward during the day and one nurse to seven patients overnight (Queensland Health 2016).

Due to the higher acuity of patients on some surgical and medical wards, such as cardiac wards, these ratios can be improved depending on ward funding. Due to the size of this particular project, two wards have been selected as the focus, namely the cardiothoracic ward and the coronary care unit. These wards service patients with serious, life-threatening heart and lung conditions. The cardiothoracic unit, being a surgical ward, has a ratio of one nurse to three patients. The coronary care unit, being a critical care area, has a ratio of one nurse to two patients.

2.4 Staffing and Rosters

Hospitals are staffed by a myriad of nurses with varying qualifications, skill levels and previous experience. Wards and specialty areas generally have a large roster of permanent nurses of varying education levels including clinical nurses (CNs), registered nurses (RNs), new graduate registered nurses (NGRN), enrolled nurses (ENs) and assistants in nursing (AINs). Permanent employees are put on a rotating roster and work shifts to cover 24 hours of the day. As permanent nurses work on the same ward daily, they are often highly skilled in their chosen speciality (Manias, Aitken, Peerson, Parker & Wong 2002).

Shortfalls in the roster can arise with vacant positions, permanent employees taking annual leave, maternity leave, emergent and sick leave. Guidelines surrounding nurse to patient ratios within Australian public hospitals means that roster gaps must be filled in order to ensure patient safety (Manias, Aitken, Peerson, Parker & Wong 2002). The perennial shortage of nurses combined with leave requirements and an increased demand for their services have contributed to a greater reliance on permanent staff working overtime and the use of agency and pool nurses (Manias, Aitken, Peerson, Parker & Wong 2002).

A pool nurse refers to an employee of the hospital who is not assigned to a specific patient care unit and is available to work in units with the greatest need. If the pool nurse supply has been depleted, roster gaps can be filled by outsourcing to nursing agencies (Manias, Aitken, Peerson, Parker & Wong 2002). Casual pool and agency nurses are greatly appreciated for their role, however as their experience and familiarity with specific specialties varies, greater responsibility and pressure often falls on the permanent employees to provide additional supervision and take up the administrative load not undertaken by short term and casual nurses (Manias, Aitken, Peerson, Parker & Wong 2002).

These issues contribute to feelings of being overworked, burn out and difficulties in attracting and retaining nurses in full-time permanent work (Manias, Aitken, Peerson, Parker & Wong 2002). To minimise the risks to the quality of patient care and to promote the best standards, health services need to have effective systems in place to ensure that all nursing staff, temporary and permanent, are appropriately qualified, experienced and fit for the roles they are asked to perform (Manias, Aitken, Peerson, Parker & Wong 2002). Effective capacity planning within a hospital relies on accurately estimating the demand for staff, services and the dynamics of patient flow (Khanna, Good & Lind 2013).

Traditionally the nursing staff assignment has been based on a volumetric approach known as the 'midnight census', that is the number of patients in the hospital at midnight (Silver & Sweberg 2013). It is used as a predictor of required staffing levels for the following day. If the facilities occupancy is low and staff numbers are high, then productivity is reduced, and the facility is at risk of losing profitability (Silver & Sweberg 2013). If the occupancy is high, taking into consideration sick calls throughout the day, excessive work for the rostered nurses can result consequently leading to hurt morale (Silver & Sweberg 2013). A flaw of the midnight census system is the underlying assumption that any point in the day, ostensibly midnight, can predict nursing care needs for patients in hospitals for the rest of the shift or following day (Silver & Sweberg 2013).

2.5 Patient-to-staff allocation process

On the ward level, the roster coordinator generally attempts to distribute an appropriate mix of CNs, RNs and ENs onto each shift for the month. This mix can be derailed on a shift-to-shift basis depending on unforeseen circumstances such as sick calls. When allocating oncoming nurses to their patients, the allocations are generally distributed as best seen fit by the nurse in charge of the previous shift or the Nurse Unit Manager (NUM) if during hours. Pool nurses are distributed to different wards by their own internal NUM based on their experience, competencies and personal preference (Saville, Griffiths, Ball & Monks 2019).

There are notable flaws in this current system. While nursing pools and agencies try to supply appropriate staff to wards, i.e., cardiac trained nurses to cardiac wards, this is not always achieved due to the overwhelming demand for pool and agency nurses (Saville, Griffiths, Ball & Monks 2019). For example, if there are minimal cardiac trained nurses within the nursing pool on a certain shift, higher acuity specialties such as the intensive care unit (ICU) or the emergency department could be allocated these nurses, leaving a skill shortage on the cardiac wards (Saville, Griffiths, Ball & Monks 2019).

This process also assumes the senior nurse on shift has a general knowledge of each staff members experience level and competencies in a specific speciality (Saville, Griffiths, Ball & Monks 2019). This is not always the case due to the huge number of staff in a hospital and their ever-changing skill set (Saville, Griffiths, Ball & Monks 2019). Other considerations include the nurse in charge being unable to make thought through decisions regarding patient nurse allocations due to immense time pressures and distractions during a shift (Saville, Griffiths, Ball & Monks 2019). It is also based on a presumed knowledge of the patient's current condition and needs. It can also be assumed that each patient's condition will not remain stable or 100% predictable during their stay or even for that day (Saville, Griffiths, Ball & Monks 2019).

The skill requirement can therefore shift dramatically in a short time. The patient's current medical status and needs combined with new admissions throughout the shift can derail even the most thought through staff allocations. As well as this, a patient requiring particularly complex care will inadvertently reduce care to other patients within that staff members allocation. This action, while not intentional, can lead to increased risk in adverse events (Saville, Griffiths, Ball & Monks 2019).

2.6 Patient-to-staff allocation risks

Appropriate nurse to patient allocations and adequate skill mix of nurses providing care is imperative for many reasons. The association of nursing skill mix with mortality, patient ratings of hospitals, the frequency and severity of adverse patient outcomes and nurse job satisfaction is well documented (Aiken, Clarke & Sloane 2002). Studies show that an imbalanced workload is an important factor in predicting turnover, burnout and dissatisfaction among nurses in the practice environment (Aiken, Clarke & Sloane 2002). Nurse turnover has recently gained greater attention due to strong correlations with patient outcomes including patient falls and infections, low staff morale, poorer job satisfaction and quality of patient care (Duffield, Roche, Blay, Thomas & Stasa 2011; Hayes et al. 2012; O'Brien-Pallas, Murphy, Shamian, Li, & Hayes 2010). Higher rates of staff turnover also place considerable demands on hospital budgets.

Although there are limited published information on nursing turnover rates and costs within Australia to date, limited studies undertaken in Queensland and the Northern Territory have reported problems attracting and retaining nurses within their hospitals (Eley, Buikstra, Plank, Hegney, & Parker 2007). Queensland public hospitals have previously reported a turnover rate of between 12% and 31.9% in the early to mid 2000's (Eley, Buikstra, Plank, Hegney, & Parker 2007). Figures from the Northern Territory show a turnover rate of 38% and a mean turnover cost of \$10,734 per commencing nurse (Eley, Buikstra, Plank, Hegney, & Parker 2007).

Research suggests that improving organizational initiatives related to work–life balance, such as staffing, and scheduling promotes job satisfaction and retention leading to cost saving within hospitals (Pabico & Graystone 2018). Along with budget demands, inappropriate nurse-patient allocations have been shown to have a direct correlation with an increase in adverse events. These range in severity and include falls, central nervous system complications, pressure ulcers, deep vein thrombosis, sepsis, ulcers and bleeds, shock, cardiac arrest, pneumonia, increased average length of stay and increased mortality rates (Twigg & Duffield 2009).

2.7 Patient risks and outcomes

Falls within Australian hospitals can cause significant harm to patients as well as having a significant impact on the cost of care. Fall-related injury is one of the leading causes of hospital-acquired morbidity and mortality in older Australians and results in pain, bruising, lacerations and fractures ultimately leading to longer hospital stays and higher costs (Cameron, Gillespie, Robertson, Murray, Hill & Cumming 2012). In many cases, falls causing harm are preventable.

Much research since the 1990s has concluded the importance of nurse staffing to patient safety. Higher numbers of RNs and appropriate RN skill mix have been linked to lower rates of inpatient falls (Cameron, Gillespie, Robertson, Murray, Hill & Cumming 2012). Researchers found that nursing units with more experienced RNs had lower total fall rates which suggests that quality of nursing care over quantity of staff can reduce incidence of falls (Dunton, Gaiewski & Klaus 2007). Factors such as nursing experience, tenure on the unit and education, as well as the ability of staff to work as a team have a positive impact on the number of inpatient falls in a unit (Cho, Ketefian & Barkauskas 2003).

Evidence suggests appropriate nursing allocations, education and experience can assist in the early detection of deterioration and life threatening infections including sepsis (Torsvik, Gustadm Mehl, Bangstad, Cinje, Damas & Solligard 2016). Sepsis is a world-wide public health issue and claims thousands of lives each year. The incidence of sepsis is escalating as the population ages, and its treatment is becoming an increasingly

significant burden on national health care expenditure (Prashant, Deven, Omender, Rohit & Vikas 2011). Sepsis costs an estimated \$846 million to treat in Australian ICUs annually. Sepsis awareness, early recognition and identification, resuscitation, referral to specialist care and prompt treatment is essential to improve survival (Prashant, Deven, Omender, Rohit & Vikas 2011).

Several recent studies conclude that continuing education is identified as an important factor in recognizing patient deterioration (Cox et al. 2006; Pantazopoulos 2012; Chua et al. 2013; McDonnell et al. 2013; Hart et al. 2014). Ongoing specific clinical education and skills training was identified as imperative in enabling nurses to recognize and respond to patient deterioration (Cox et al. 2006; McDonnell et al. 2013). The level of education was identified as a significant predictor in ward nurses' ability to promptly recognize the onset of sepsis (Pantazopoulos 2012). Registered nurses who had graduated from a 3-year university education identified patient deterioration significantly quicker than enrolled nurses who had graduated from a 2-year educational programme (Pantazopoulos 2012). Registered nurses who had obtained a postgraduate qualification were more self-confident in recognizing patient deterioration (Pantazopoulos 2012). It was therefore determined that allocating a nurse with a greater experience level to a patient that is acutely unwell or at risk of deteriorating can improve a deteriorating patients' outcomes (Prashant, Deven, Omender, Rohit & Vikas 2011).

2.8 Chapter Summary

This literature review provides evidence that well planned patient-to-staff allocations can lead to improved patient outcomes, staff satisfaction and help reduces costs in an acute care facility in Australia. The literature review did not find any supporting evidence that NLP or AI technology has been implemented or developed for patient-to-staff allocation. This gap in the literature provides strong reasoning for this project to progress and introduce technology that will aim to improve patient outcomes and staff satisfaction.

Chapter 3. Natural Language Processing

3.1 Chapter Overview

To develop the scope of the project and define appropriate techniques for the proposed application an in-depth investigation into Natural Language Processing, its techniques and approaches has been completed. This research included product release notes and examples, papers and applications of natural language processing. To ensure this research dataset was not too broad only NLP applications in Python were reviewed. Finally, a selection of techniques and approaches are defined based on the research.

3.2 Python programming language

The Python programming language is a high-level programming language and was designed based on a small core with the ability to be extended by adding modules. This approach has seen the programming language grow in popularity and has been chosen to complete this project predominantly due to its simple syntax and large number of NLP and machine learning libraries. Python is also an OSI-approved open-source language, making it freely usable and distributable even for commercial use. Additionally, Python's growing popularity and use in engineering applications makes it a good choice for the project and any future works.

3.3 Introduction to Natural Language Processing (NLP)

Natural Language Processing (NLP) is a field of Artificial Intelligence that provides machines or computers the ability to read, listen, understand, and derive meaning from human or natural languages. The human language is generally classified as unstructured data, this unstructured data can be difficult to manipulate and does not neatly fit into forms that are easily understood by machines, such as relational databases. By processing this unstructured data using any number of combination of techniques and algorithms it is possible to process the unstructured data into structured meaningful data that allows machines to extract critical information.

Linguistics analysis by humans is not just to understand words based on their definition but also its context and sentiment, therefore NLP must also consider this in its process and understanding. NLP can and is often multilayered using many techniques to deal with sentiment and context to deliver a specific outcome. The following techniques have been researched and tested in the process of developing this project, it should be noted that the following techniques have been considered with relation to the English language only.

3.4 Data Cleaning

The first step in NLP is to clean the data into a format that a machine can interpret and understand. This data cleaning is completed over a series of steps and techniques of which the following outlines the most common practices used to clean text-based data for the purpose of NLP.

The removal of punctuation such as commas, full stops, uppercase etc, helps to prevent any confusion to the machine when identifying line breaks or periods. The data is then broken down into individual words or sentences and stored as a list or doc type object in a process known as tokenization. This list of words can then be compared to a selection of "stop" words which would be removed from the data leaving only unique and relevant words to assess.

For the case of neural networks, it is not possible to process raw data, like text files, encoded JPEG image files, or CSV files. They process vectorized & standardized representations. Text files need to be read into string tensors, then split into words. Finally, the words need to be indexed & turned into integer tensors.

3.5 Punctuation

Punctuation does not offer any value or meaning to the NLP model and if left in can result in inaccurate or misleading results therefore, the removal of punctuation such as commas, full stops, quotations etc is generally the first step to cleaning the data. Often it is considered good practice to also reduce all letters to lower case, except in the instance of Named Entity Recognition (NER) where the use of proper nouns is required. NER will be discussed further in a later section.

3.6 Tokenization

This technique segments text into sentences and words or tokens by breaking down a section of text by identifying a token as a sequence of characters separated by a blank space. This process can also help remove punctuation as well, further cleaning the data. By tokenizing the text, the data is converted from a single string type element to a list of words or sentences, this is an important data type in NLP as it allows iteration over the data. Research into the use of the tokenization technique did return some comment that while powerful it can pose problems based on the text being examined, in the case of medical related articles that can often have many hyphens, parentheses and other punctuation marks, removing such punctuation can cause the text to lose its true meaning.

3.7 Stop Words Removal

This technique compares the cleaned, tokenized data to a predefined list of words that can be considered to offer no value to the NLP objective. Common pronouns and prepositions such as “and”, “the” or “to” are generally included in default stop word lists, however, the stop word list can be customized to suit the application. By removing this list of words from the data it is possible to improve processing time and free up database space. The stop word list should be designed to suit the project and more modern trends suggest this technique has become less common as the removal of words can alter the context or provide inaccurate results when performing sentiment analysis.

3.8 Lemmatization

Lemmatization is the process of reducing words to their base form and grouping words of the same type together. For example, “worst” is changed to “bad”, verbs in past tense can be replaced with present tense and therefore standardize words to their root form. Lemmatization also takes into consideration the context of the word to help avoid disambiguation, this allows the process to decipher the meaning of two words which present in the same way. For example, “fan” which corresponds to both a supporter and a device used to create a current of air. By providing a part-of-speech parameter to a word, such as a noun, verb, etc. It is possible to define the context and meaning for that word and remove disambiguation.

“Lemmatization resolves words to their dictionary form (known as lemma) for which it requires detailed dictionaries in which the algorithm can look into and link words to their corresponding lemmas.” (Diego Lopez Yse 2019)

3.9 Part of Speech Tagging

Part of speech tagging is the process of defining and tagging the grammatical role of a particular word in a sentence. This allows words to be grouped into categories based on their use. This gives the machine the structured data it requires to derive insights from common nouns or which adjectives are used for a particular noun or verb.

For example, the sentence “Daniel is completing his dissertation on NLP” would provide a result of each worded tagged as follows.

Daniel	- NNP noun, proper singular
is	- VBZ verb, 3rd person singular present
completing	- VBG verb, gerund or present participle
his	- PRP\$ pronoun, possessive
dissertation	- NN noun, singular or mass
on	- IN conjunction, subordinating or preposition
NLP	- NNP noun, proper singular

3.10 Term Frequency-Inverse Document Frequency (TF-IDF)

Term Frequency-Inverse Document Frequency is the statistical analysis of how relevant a word is in a document and over a corpus. The tf-idf value of a word grows in proportion with number of times the word presents in a document (term frequency), this frequency is then offset based on the number of documents the word appears in (Inverse Document Frequency). The inverse document frequency helps to adjust the weight of the word by allowing for the possibility that some words may exist more frequently in general. Variations of the calculation of the term frequency has been developed to allow for variations in the text to help reduce any bias results.

Term frequency, $tf(t,d)$, is the frequency of term t .

$$tf(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

Where $f_{t,d}$ is the raw count of a term in a document or the number of times that term t occurs in document d .

Inverse document frequency is a measure of how much information the word provides or If it is common or rare across all documents. It is the logarithmically scaled inverse fraction of the documents that contain the word (obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient):

$$idf(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

Where:

N : total number of documents in the corpus $N = |D|$

$|\{d \in D: t \in d\}|$: number of documents where the term t appears (i.e. $tf(t,d) \neq 0$). If the term is not in the corpus, this will lead to a division by zero. Therefore, the denominator is adjusted to $1 + |\{d \in D: t \in d\}|$.

Term frequency-Inverse document frequency is then calculated as,

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D)$$

A term with a high frequency (in a given document) and a low document frequency across the corpus will result in a high weight return from the tf-idf algorithm.

Tf-idf can be implemented in python using scikit learn as follows,

```
from sklearn.feature_extraction.text import TfidfVectorizer
```

3.11 Non-negative Matrix Factorization

Nonnegative matrix factorization (NMF) approximates a nonnegative matrix by the product of two low-rank nonnegative matrices. Since it gives semantically meaningful result that is easily interpretable in clustering applications, NMF has been widely used as a clustering method especially for document data, and as a topic modelling method.

When multiplying matrices, the dimensions of the factor matrices may be significantly lower than those of the product matrix and it is this property that forms the basis of NMF. NMF generates factors with significantly reduced dimensions compared to the original matrix. For example, if \mathbf{V} is an $m \times n$ matrix, \mathbf{W} is an $m \times p$ matrix, and \mathbf{H} is a $p \times n$ matrix then p can be significantly less than both m and n .

NMF used for NLP, if we consider \mathbf{V} to be the term document matrix, each row of the matrix \mathbf{H} is a word embedding and each column of the matrix \mathbf{W} represents the weight of each word in each sentence (semantic relation of words with each sentence). It is assumed that all the entries of \mathbf{W} and \mathbf{H} are positive as all the entries for \mathbf{V} are positive.

$$\begin{bmatrix} \text{W} \\ \text{W} \\ \text{W} \end{bmatrix} \times \begin{bmatrix} \text{H} \\ \text{H} \\ \text{H} \\ \text{H} \\ \text{H} \end{bmatrix} \approx \begin{bmatrix} \text{V} \\ \text{V} \\ \text{V} \\ \text{V} \\ \text{V} \end{bmatrix}$$

NMF can be implemented in python with the use scikit learn libraries using the following code,

```
from sklearn.decomposition import NMF
```

From scikit learn NMF is implemented based on the following equations

$$\begin{aligned}
& 0.5 * ||X - WH||_{loss}^2 + alpha * l1_{ratio} * ||vec(W)||_1 \\
& \quad + alpha * l1_{ratio} * ||vec(H)||_1 \\
& + 0.5 * alpha * (1 - l1_{ratio}) * ||W||_{Fro}^2 \\
& + 0.5 * alpha * (1 - l1_{ratio}) * ||H||_{Fro}^2
\end{aligned}$$

Where:

$$||A||_{Fro}^2 = \sum_{i,j} A_{i,j}^2 \text{ (Frobenius norm)}$$

$$||vec(A)||_1 = \sum_{i,j} abs(A_{i,j}) \text{ (Elementwise L1 norm)}$$

The NMF model can be manipulated for improved results based on the requirements of the model and the data set. Parameters such as the beta-loss can be selected based on divergence calculation ('frobenius', 'kullback-leibler', 'itakura-saito') or numerical solver (Coordinated Descent, Multiplicative Update).

3.12 Tensorflow

Tensorflow is an open-source Machine Learning library developed by Google. It provides access to a range of tasks focusing predominantly on training and inference of deep neural networks. Often referred to as an infrastructure layer for differentiable programming. Differentiable programming functions as a numeric computer program that can complete differentiation via automatic differentiation, because of this it is possible to complete optimization based on gradient of parameters, often via gradient descent. Tensorflow provides 4 main abilities:

- Efficiently executing low-level tensor operations on CPU, GPU, or TPU.
- Computing the gradient of arbitrary differentiable expressions.
- Scaling computation to many devices, such as clusters of hundreds of GPUs.
- Exporting programs ("graphs") to external runtimes such as servers, browsers, mobile and embedded devices.

While these main abilities are outside the scope of this project it is relevant for any further work should the project progress beyond this initial scope. Tensorflow is available as a stable release for python and was a natural choice for investigation and testing with respect to the project. Tensorflow also provides a stable API with Keras

3.13 Keras

As mentioned in the previous section, Keras is a deep learning API written in python and operates on top of Tensorflow. Keras was developed with a focus on enabling fast experimentation and taking ideas to functional results as fast as possible. Keras implements the neural-network building blocks such as layers, objectives, activation functions, optimizers and several other tools designed to make working with text and image data easier. Offering simple and easy to develop sequential models by stacking layers or providing extended control over unique or exotic cases through subclassing. The core data structures of Keras are Layers and Models, the simplest of which is a sequential model or a linear stack of layers and can easily be added to the model as in the example below,

```

from tensorflow.keras.layers import Dense

model.add(Dense(units=64, activation='relu'))
model.add(Dense(units=10, activation='softmax'))

```

3.14 Multiclass text classification with Tensorflow and Keras

Text classification is a machine learning technique that assigns a set of predefined categories and or classes to text data. As an example, the following sentence can be analysed by a text classifier and its content assigned relevant tags/labels, such as *Respiratory sound*, and *reduced*.

“Chest auscultated. Diminished sounds in the bases bilaterally”

Text classification can be achieved manually or automatically, for this project only automatic machine learning based classification has been considered. Machine learning text classification can learn to make text classification based on past observations, so it is critical to this approach that suitable levels of training data is available. There are many text classification algorithms some of the most popular include Naïve Bays family of algorithms, support vector machines (SVM) and deep learning. Due to the complexity of the data and large number of possible text classifications only the deep learning method has been chosen and tested.

There are two main learning architectures for deep learning algorithms, Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN).

Convolutional Neural Networks extract features from an image and convert that feature to a smaller dimension without losing the characteristics of the original feature. CNN is commonly used for image recognition in applications such as face recognition and image classification, for this reason it will not be considered in this project.

Recurrent Neural Networks recognise a data sets sequential characteristics, maintain a memory based on history information and use patterns to predict the next data point. For this reason, RNN is commonly used in speech recognition and NLP due to the sequential nature of text or language.

3.15 LSTM – Long Short-Term Memory

Long Short-Term Memory or LSTM is a widely used deep learning algorithm used for recurrent neural networks. LSTM provides feedback connections to layers within the neural network and allow for information to persist. Recurrent neural networks (RNN) can not remember long term dependencies due to a well documented problem known as vanishing gradient. LSTMs are specifically designed to avoid the vanishing gradient and provide a solution to long-term dependency problems.

The LSTM algorithm consists of a Forget Gate, Input Gate and Output Gate, each section of the cell updates information based on the previous timestamp. The Forget Gate determines if the data or information from the previous time stamp should be forgotten. The Input Gate is used to identify the criticality of the new information delivered by the input. The Output Gate must provide a prediction or output based on the information provided by the two earlier function gates.

The functions outlined above provide a short overview of a sequential LSTM model that can be implemented as part of a complete deep learning model using Tensorflow and Keras.

3.16 SpaCy

spaCy is another open-source library for advanced natural language processing available in python. spaCy's main focus is to provide NLP for production usage rather than teaching and or research as is the case with other libraries such as NLTK. spaCy supports deep learning algorithms including features to support neural network models for part of speech tagging, dependency parsing, text categorization and named entity recognition (NER). spaCy provides several free online training and development resources so it was a good choice for the project.

3.17 SpaCy – Named Entity Recognition (NER)

Named Entity Recognition (NER) is the process of assigning a name to a “real-world object”, for example a person, company, product etc. For this project competency or “COMP” has been chosen as the entity and is assigned to unigrams, bigrams, or trigrams in line with key word indicators outlined by the clinical resource. These will be outlined further in Chapter 4. Entities provide several pieces of information to help define their use and importance. When an entity is recognised by the model it assigns an entity label, in this case COMP as well as the start position of the first character and end position of the last character of the entity. In the example below it can be seen that “hypotension” has been assigned the entity “COMP” as well as the start index and end index within the sentence token.

```
"borderline hypotension as per iEMR",
{
  "entities": [
    [
      11,
      22,
      "COMP"
    ],
  ],
}
```

3.18 Chapter Summary

This chapter provided an introduction to NLP techniques and how these might be applied to text or natural language data. Furthermore the 4 approaches chosen for testing and validation in this project were introduced. TF-IDF and TF-IDF + NMF were chosen for a topic classification based approach to data extraction. SpaCY implements an NLP model for custom named entity recognition and Multiclass text classification makes use of a recurrent neural network provided by Tensorflow and Keras.

Chapter 4. Data Requirements and Manipulation

4.1 Chapter Overview

A critical step in the design and implementation of an NLP model is collecting and organising the data in the correct format. This chapter outlines the data sets required the formats in which they exist and how they will be used. A main focus exist around collecting and manipulating the data from which information is to be extracted.

To process natural language using NLP algorithms and ML models it is critical that the text data is managed and moulded into a state that a machine can understand

4.2 Data Type

As outlined in earlier chapters, this project aims to match a staff member (nurse) to a patient. Matching is based on a patient's acuity and required level of care to a staff members experience and ward specific acquired skills. For this matching to occur three data sets must be available for analysis.

1. Nurse progress notes (patient to be assigned)
2. Key word/phrases for data extraction
3. Staff experience and Competencies

4.2.1 Nurse Progress Note

Progress notes are completed by a registered nurse at the end of each shift as an update on the patient's recovery. The nurse progress note contains a detailed outline of the treatments given, what may be required in the future, the current state and progression of the medical condition for which the patient was admitted to hospital and any other relevant information that may be useful to oncoming nurses, this may include mental state or mobility as an example. Progress notes have been a critical piece of patient recovery and care for many years, however, until recently these notes were completed and stored manually, handwritten. With improvement to technology and a need for sustainable practice many large facilities and organisations have moved to digital records, which includes the nurse progress note. It is this shift to digital information and recording that provides the necessary data required for this project.

Figure 1 shows an example of a nurse progress note from a cardiac unit based in a Southeast Queensland Hospital. Nurse progress note do have a very vague expected structure and specific details that should be included, however, each nurses articulation, grammar and general structure will vary. It was hoped that this would be taken into account in this project but unfortunately a lack of access to data has not allowed for this to occur. The 10 notes used for testing and data generation can be found in appendix B.

Received care of patient at 15:30hrs from Lismore Base Hospital.
Thrombolysed at 13:11hrs in LBH.

Neuro

GCS 15. Alert and orientated.
2/24 neurovascular obs attended. PEARL.

Cardiac

ECG attended on arrival to ward.
ST elevation still present but resolving in V1-4.
Noted TWI in inferior leads.
Telemetry monitoring in SR.
Nil arrhythmias noted ATOR.
BP remains stable.
Nil c/o chest pain.

Resp

SpO₂ 95% RA.
Nil respiratory distress.

Gastro

Tolerating diet and fluids.
BSL 20.2. Given STAT Insulin as per CCU Reg.
Ketones 1. Will continue to monitor.
BNO.
Nil c/o nausea.

Renal

Passing urine in the bottle independently.

Lines

PIVC R) CF patent and flushed.
PIVC L) CF patent and flushed.
Routine bloods taken on arrival.

Figure 1:Nurse progress note

4.2.2 Competency key word(s)

Certain key words and phrases have been defined to allow for specific data extraction to take place in the varying NLP algorithm scenarios. The key word or phrase list has been supplied by the clinical resource based on the 10 example notes used. The list identifies specific terms that can be used to determine a required competency for a patient's care needs.

The staff competency list has been defined in separate json files each with a list of key phrases in the form of unigrams, bigrams or trigrams. The json file format provides an easy format for which to iterate across and compare extracted patient acuity to staff competencies. The following json file examples depict the file structure and use line 0 as the title of the competency, this is later used to identify which competency the key word belongs to and match to the Nurse class attributes containing competencies.

12 Lead ECG and interpretation – 12lead.json

```
[
  "12 lead ECG",
  "telemetry",
  "sinus rhythm",
  "SR"
]
```

Arterial Sheath Removal – ArterialSheath.json

```
[
  "Arterial Sheath Removal",
  "arterial",
  "Arterial",
  "sheath",
  "arterial sheath",
  "femoral",
  "femoral site"
]
```

The full list of competency json files can be reviewed in appendix C.

4.2.3 Staff Experience and Competencies

Each ward in an acute care facility has a specific set of unique competencies that are required for the care of the patients admitted to that ward. In the case of this project a Cardiothoracic Ward and a Coronary Care ward have been chosen from the Gold Coast University Hospital (GCUH). The cardiothoracic ward involves caring for patients pre and post heart and lung surgery. The coronary care unit (CCU) involves caring for patients post heart attacks, managing arrhythmias and post interventional procedures. The staff competencies have been taken directly from the requirements for working in these 2 wards at GCUH. When a staff member starts on the ward, they immediately begin to work towards completing these tasks, which can take 3 to 18 months depending on the frequency of the task in the ward.

Original plans for the project aimed to include staff from the two wards at GCUH as direct participants in the project however, time constraints and increased restrictions relating to staff information and involvement based on ethical considerations meant that staff participation was not viable for this project. It was decided that staff members would be hypothetical based on the competencies, education, and experience.

A list of the competencies is shown in table 1:

Table 1: Staff competency table

Education level	Previous experience	Competencies
Assistant in Nursing (AIN)	Nursing home	12 lead ECG and interpretation
Enrolled Nurse (EN)	Medical/surgical nursing (general wards)	Chest pain management
New graduate nurse (NGRN)	Mental health	Basic life support
Registered Nurse (RN)	Palliative	Chest auscultation
Senior Registered Nurse (RN1)	Critical care nursing (ICU/ED/CCU)	Managing a deteriorating patient
Clinical Nurse (CN)	Ventilation and Bipap experience	Wound management
	Team leading skills	Underwater seal drain management and removal
	Educating and supporting of junior nurses	Epicardial and temporary pacing initiation and management
	Other - Nursing	Pacing wire removal
		Advanced life support
		Cardiac advanced life support (reopening sternotomies)
		Blood sampling and cannulation
		Arterial sheath removal
		TR band management
		STEMI management
		Thrombolysis management
		Inotropic management
		Epidural and Regional block pain management

An example staff member entered as a child of class "Nurse" is shown in the code example below,

```
n1 = Nurse("1", "Registered Nurse", "Nursing Home", "Critical Care", "", "Wound Management",
          "Basic Life Support", "Chest Pain Management", "STEMI Management")
n1.newNurse()
```

Program output:

New Nurse added

```
{'ID': '1', 'Education': 'Registered Nurse', 'experience1': 'Nursing Home', 'experience2': 'Critical Care',
'experience3': '', 'comp1': 'Wound Management', 'comp2': 'Basic Life Support', 'comp3': 'Chest Pain
Management', 'comp4': 'STEMI Management'}
```

4.3 Data Collection

The aim of this project is to look specifically at the unique medical recovery journey of a patient in acute care and provide the appropriate level of care relevant for the patient at each point in their recovery journey. To achieve this the collection and organisation of data is critical to the development, testing, and overall success of the project. It was identified in the early stages of the project that a significant amount of real-world data would be the preferred data set to complete the project. A large real world data set would provide a suitable level of training, testing and validation to take place. The data, while considered low risk is still personal and private information relating to an individual's health and for this reason it was a requirement of both USQ and the Gold Coast University Hospital that an ethics application was completed to ensure the protection and ethical use of sensitive data. A project outline was submitted based on the project progress report submitted as part ENG4112.

The original request for data had assumed that the progress notes could be extracted as a report or bulk file selection, unfortunately this part of the digital records is not reportable and was required to be extracted manually. To ensure that no unexpected identifying data exists in the notes a name search was proposed to be completed on the final corpus of notes to ensure no identifying data remained.

GCH research approved the projects ethics application and the request for data moved to the data custodians (medical records), unfortunately the request for data retrieval continues to wait for approval based on a request for further information around the physical retrieval of data. Due to time constraints and the significant amount of work to review, classify and test the data it has not been included in the final submission of the project.

The GCUH ethics application and USQ application can be viewed in appendix E and F respectively.

4.3.2 Developing hypothetical data for multiclass text classification

Due to significant delays in approval for the retrieval of data from the Gold Coast University hospital subsequent removal of real data from the project an alternate solution to produce a data set has been developed. 10 progress notes have been written by the clinical resource covering several of the outlined nurse competencies. To develop the required amount of data, a data generation function has been written and completes the following tasks for each of the 10 notes,

1. Selects a word at random every 1 to 10 words in the progress note.
2. Checks that the selected word is not a key word related to the competency.
3. Replaces the selected word with a random word from a medical corpus (approx 100,000 words).
4. Repeats until the end of the progress note.
5. Saves the new note as a .txt file

Initial testing looked at 100 progress notes per competency however, this was later increased to 1000 notes per competency for the multiclass text classification model. The large number of data sets is only required for the training, testing and validation of the multiclass text classification model.

4.4 Progress Note Data

The following table shows the 10 example notes used in the testing, validation and basis of the hypothetical generated data. Each note has the competencies listed key word(s) for extraction and a manual (expected) count of number of key phases for each competency.

Table 2: Progress notes required competencies

Progress Note	Competencies required	Key words	Number of key phrases
1	Wound management	Infected, Wound, vac dressing, exudate,	5
	Basic life support	Oxygen requirement, increasing oxygen requirement, O2, 2L O2, Hypotension	2
	Blood glucose monitoring	BGL, blood glucose level, blood sugar	1
	Blood sampling and cannulation	Blood cultures, BCs	1
	PICC line management	PICC line	2
2	12 lead ECG interpretation	Telemetry, sinus rhythm, SR	2
	TR band management	TR Band	2
	Arterial sheath removal	Femoral site	3
3	Arterial sheath removal	Arterial sheath	1
	12 lead ECG interpretation	Telemetry	2
4	12 lead ECG interpretation	Telemetry, sinus rhythm, SR	2
	Wound management	skin tear	1
5	Epidural and Regional block pain management	Fentanyl PCA, PCA, Naloxone	4
	Epicardial pacing	pacing box, VVI, PR	2
	Wound management	Wound, comfeel dressing, dressing, alevyn dressing	9
	12 lead ECG interpretation	telemetry	1
6	12 lead ECG interpretation	Telemetry, sinus rhythm, SR	3
	thrombolysis management	lysis, thrombolysed, lysis protocol	1
7	Epidural and Regional block pain management	Fentanyl PCA, PCA, Naloxone	3
	12 lead ECG interpretation	telemetry	1
	Epicardial pacing	pacing box, VVI, PR	3
8	12 lead ECG interpretation	Telemetry, SR, ECG,	3
	STEMI management	STEMI	1

	thrombolysis management	lysis, thrombolysed, lysis protocol	2
9	STEMI management	Cardiac Catheter Lab, Percutaneous Coronary Intervention, Left Anterior Descending, PCI.	3
	12 lead ECG interpretation	Telemetry, SR	2
	TR band management	TR band	1
10	Inotropic management	complete heart block, second degree heart block, isoprenaline	4
	12 lead ECG interpretation	Telemetry, sinus rhythm, SR	1

4.5 Chapter Summary

This chapter provided a data organisation plan and identified the types of data required to achieve the projects goals. It also identified how data may be stored and used with in the models. A detailed account of attempts made to retrieve real world data have been discussed and an appropriate work around has been outlined.

Chapter 5. System Layout and Testing Scenarios

5.1 Chapter Overview

The previous chapters have outlined the technical theory for information extraction and importance of data management to achieve this. Chapter 5 provide a general overall system design to which each model will be tested against. Global functions such as staff allocation and competency identification are defined. Each of the models are detailed in a testing scenario.

5.2 General System Layout

The general system layout has been defined to keep the number of variables between NLP models to a minimum and provide a level of control to the testing. Each model loads nurse progress note(s) depending on the model architecture one or many notes may be required for training purposes, each model then cleans the data of punctuation and stop words and formats the data into tokens (words or sentences model dependent) and then compares the NLP object with the staff competency key word files to produce a competency or set of competency key words. The results of the NLP models are then used with the staff.py allocation function to determine if the correct competencies and staff have been identified. During the testing and development phase the key word list return was used to validate against key words to help with development efficiency. Algorithms and parameters are then adjusted to improve accuracy and reliability.

Some of the machine learning models require training and validation data prior to testing and the details and results of this validation will be outlined as part of the test scenario.

5.3 Staff function

The staff function has been developed to be as simple as possible to ensure it can be reused for all models being tested. Staff competency data is entered and stored via a function that implements a Class "Nurse" comprising of class parameters such as staff ID,, education, 2 previous experiences and 6 competencies. The staff.py program file also includes a keyword matching function and allocation algorithm. The key word results of each model are passed as a list variable for each patient/note tested, this list variable is then compared to all competency key word json files and results stored in a competency required variable. The comp_req variable is then used to compare against a staff members (class Nurse) attributes performing a final allocation or returning a warning of recommendation that the staff member does not have the required competencies. All matching details are then saved as a patient and staff json file.

The 3 primary functions of staff.py,

- Hypothetical staff members class defined.
 - Defines education, 2 experiences and 6 available competencies
- Match results of the NLP to key words for predefined competencies.
- Allocated a patient to a staff member based on competencies found in NLP modelling with competencies entered as class attribute.

5.3 Testing Scenarios

Aim: The aim for all models tested is to accurately extract the necessary key word(s) from the raw progress notes and save the results in a list or NLP object variable that can then be passed to the staff function to complete the allocation process and determine the accuracy of the model.

The goal for each model was to reach a key term extraction of 75% or greater at which point the model results could be passed to the allocation program and competency matching accuracy recorded.

For each of the model testing scenarios the following parameters have been defined.

Progress notes tested : 10

Predetermined competencies required for patient care:

- Wound Management
- 12 Lead ECG
- PICC Line Management
- Arterial Sheath Removal
- Basic Life Support
- TR Band Management
- Blood glucose monitoring
- Blood sampling and cannulation
- Epidural and Regional block pain management
- Epicardial pacing
- Thrombolysis Management
- STEMI management

Number of staff available: 4 (with varying competencies)

5.3.1 Model 1 – Topic Modelling tf-idf

Python model: Model_1_tf-idf.py

Description: using the term frequency-inverse document frequency method described in chapter 3, a test model was developed under the following parameters,

- Max Features = 40 - maximum the number of results found per Note
- Max_df = 1.0 - maximum number of documents the term exists in
- Min_Df = 0.0 - minimum number of documents the term exists in
- Ngram_range = (1, 3) – Terms can be found as unigram, bigram or trigram

The configuration of the TF-IDF vectorizer can be considered to be operating in reverse of its intended use. Due to the need for the program to search an individual note for feature extraction, the max_df and min_df are set to parameter extremes.

5.3.2 Model 2 – Topic Modelling with tf-idf and NMF

Python model: NMF.py

Description: The NMF model extends the TF-IDF model outlined above, the following additional parameters have been added to the NMF feature in sklearn,

NMF Model configuration

- n_components = 4
- Solver = “mu” – Multiplicative update solver
- Beta_loss = ‘frobenius’

TF-IDF updates

- Max_features = 40 and decreasing by 10 until <= 10

Multiplicative update solver or weight method is an algorithm commonly used for decision making and prediction. In its most basic design, the solver consists of having two experts at equal weights, the solver then updates the weights multiplicatively and iteratively according to feedback of how well the expert has performed.

5.3.3 Model 3 – Multiclass text classification with Keras and TesnorFlow

Python model: Multiclass.py

Description: The multi-class text classification program follows the same initial principles of data cleaning as outlined in previous chapters it then proceeds to implement a stochastic gradient descent, an iterative learning algorithm that uses a training dataset to update a model.

Model data is entered in 2 formats,

1. A csv file containing the competencies as column headers and data entered as either a 1 or a 0 indicating if that competency exists in a progress note. This data is then loaded into the model as labels and training data using a panda data frame.
2. Progress notes are loaded as .txt files and combined in a list format

To prepare the data it is first split into training and test sets followed by tokenisation of the progress note and padding of sequences to ensure all data segments are of equal size. A standard glove file is used to provide word embeddings. The deep learning model is then created with 3 layers. Initial testing was completed around 3 outputs or classes, this was later increased to 6.

1. Embedding layer – The model accepts an integer matrix of size (batch, input length)
2. LSTM layer – accepts the embedding layer as an input
3. Dense layer (output) – Provides 3 outputs based on input from the LSTM layer
 - a. The 3 outputs theoretically represent the prediction of the model, i.e., the model predicts how likely an unseen progress note may be of a specific class (competency).
 - b. 6 outputs of class type "12 Lead ECG", "Basic Life Support", "Arterial Sheath Removal", "Wound Management", "Inotropic Management", "TR Band Management"

The following hyper parameters have been adjusted and tested in an attempt to optimize the model,

- LSTM_Layer_1 = 64, 128 and 256
- Batch size = 8, 16, 32
- Epochs = 4, 8 and 16
- Validation split = 0.2

The batch size is a hyperparameter of gradient descent that controls the number of training samples to work through before the model's internal parameters are updated.

The number of epochs is a hyperparameter of gradient descent that controls the number of complete passes through the training dataset.

5.3.4 Model 4 – Named Entity Recognition with Spacy

Python Model: Spacy_train.py, SpacyModel.py

Description: The Spacy Named Entity Recognition (NER) model has been split into 2 parts, part A creates and trains a NER model, while part B validates the model by testing unseen data (progress notes) and calls the staff.py program for the allocation process. Figure 2 shows a graphic representation of a Spacy model workflow.

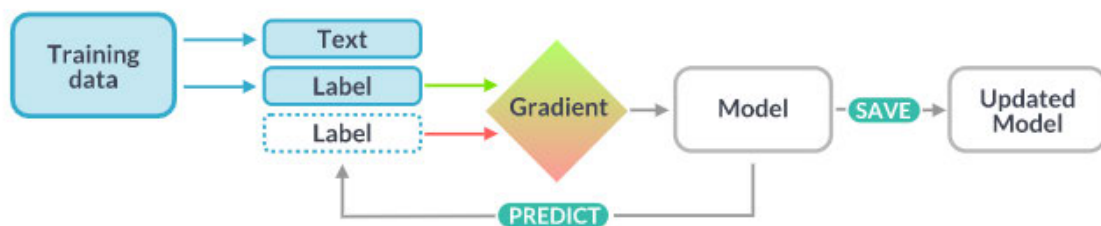


Figure 2:Spacy model layout

Part A: To train a custom NER model in Spacy 3.0 a config.conf file is required based on the application of the model being designed. A base config file can be downloaded from the Spacy website, with some general selection criteria available, for this project the following base config parameters were selected.

Language = English

Components = ner

Hardware = CPU

Optimize for = accuracy

The default config has been used with the exception of a required update to [paths] for location of spacy binary file on the local machine. To configure the Spacy NER model a custom NLP pipeline must be added, the pipeline must then be fed an entity label and pattern based on defined keywords as outlined below.

```
for item in data:
    pattern = {
        "label": type,
        "pattern": item
    }
```

Where the “data” variable contains the contents of a json file listing the key word entities and the item variable contains the entity label, in this instance “COMP”. The full list of key words can be found as appendix D the NLP model is then saved to disk.

Part B: Once the model has been created and trained using a training data set it can now be tested against unseen data for accuracy. To achieve this the NER model is loaded into the python program as a NLP object. An unknown progress note is then also loaded, and the text cleaned in the same format as the training data. The progress note is then converted to a doc object by running data through the loaded NLP model. If a keyword is identified in the progress note this is saved as a list variable and passed to the staff.py allocation functions.

5.4 Chapter Summary

This chapter gave a detailed account of the methodology behind the testing and validation of each model. It also described the configurable parameters and how these impact the results of the model. All models use the staff.py program to verify results.

Chapter 6. Evaluation and Testing

6.1 Chapter Overview

The results and analysis of the NLP and Machine learning models are presented in this chapter. Results are first presented in a json file format showing the key words identified, associated competencies and the staff member for which the competencies have been compared. Finally, the accuracy of each model is shown in a series of tables.

6.2 Model 1 – Topic Modelling with TF-IDF

The results of the TF-IDF model have been saved to a json file for each patient progress note per staff member available, examples can be seen below. The json files show the keywords found by the TF-IDF algorithm, the associated competencies identified (if any), the staff member defined competencies and if that staff member can provide appropriate care for the patient based on the competencies listed.

Nurse Progress Note 10, Staff Member 1 results,

The below json file output for “patient” 10 and staff member 1 shows the key phrases found in the progress note, the associated competencies and staff competencies, it can be seen by review that a match was achieved and the file has returned a false result for appropriate staff member.

Key words found in progress note

```
[
  [
    "achieved",
    "aiming",
    "block",
    "bpm",
    "heart",
    "heart block",
    "isoprenaline",
    "mlshr",
    "mlshr lines",
    "mlshr lines pivc",
    "monitoring",
    "nil",
    "nil pain cardiac",
    "nil respiratory",
    "nil respiratory distress",
    "orientated",
    "orientated nil",
    "orientated nil pain",
    "pain",
    "pain cardiac",
    "pain cardiac telemetry",
    "passing",
    "passing mlshr",
    "passing mlshr lines",
```

```

    "patent",
    "patent flushed",
    "patent flushed pivc",
    "patent running",
    "patent running isoprenaline",
    "patient",
    "patient consented",
    "patient consented ppm",
    "patient hrs",
    "pivc",
    "ppm",
    "remains",
    "sp ra nil",
    "team resp sp",
    "telemetry",
    "telemetry monitoring"
  ]
]
Patient care requirements
[
  "Inotropic Management",
  "Inotropic Management",
  "12 Lead ECG"
]

Staff competencies
[
  "",
  "Wound Management",
  "Basic Life Support",
  "Chest Pain Management",
  "STEMI Management"
]
Staff has appropriate competencies for Patient = false

```

Nurse Progress Note 10, Staff Member 3 Results,

Key phrases and associated competencies remain the same as previous note however, the patient requirements have been compared to staff member 2 who holds the correct competencies for this patient and therefore the system has returned a True result for allocation of patient to staff.

Key words found in progress note

```

[
  [
    "achieved",
    "aiming",
    "block",
    "bpm",
    "heart",
    "heart block",
    "isoprenaline",
    "mlshr",
    "mlshr lines",
    "mlshr lines pivc",
    "monitoring",
    "nil",

```

```

"nil pain cardiac",
"nil respiratory",
"nil respiratory distress",
"orientated",
"orientated nil",
"orientated nil pain",
"pain",
"pain cardiac",
"pain cardiac telemetry",
"passing",
"passing mlshr",
"passing mlshr lines",
"patent",
"patent flushed",
"patent flushed pivc",
"patent running",
"patent running isoprenaline",
"patient",
"patient consented",
"patient consented ppm",
"patient hrs",
"pivc",
"ppm",
"remains",
"sp ra nil",
"team resp sp",
"telemetry",
"telemetry monitoring"
]
]
Patient care requirements
[
  "Inotropic Management",
  "Inotropic Management",
  "12 Lead ECG"
]

Staff competencies
[
  "TR Band",
  "Advanced Life Support",
  "12 Lead ECG",
  "PICC Line",
  "Inotropic Management"
]
Staff has appropriate competencies for Patient = true

```

Nurse Progress Note 6, Any Staff Member Results,

Results reported for progress note 6 show a failure to identify any competency key phrases and several empty tokens. Correct key phrases and required competencies can be reviewed in Table 2.

Key words found in progress note

```

[
  [
    "hrs",
    "patient",

```

```

"patient hrs",
"patient hrs lismore",
"received",
"received care",
"received care patient"
],
[
"alert",
"arrival",
"attended",
"neuro gcs alert",
"neurovascular",
"neurovascular obs",
"neurovascular obs attended",
"nil",
"nil arrythmias",
"nil arrythmias noted",
"nil chest",
"noted",
"pearl",
"pearl cardiac",
"pearl cardiac ecg",
"present",
"present resolving",
"present resolving noted"
],
[
"nil",
"ra",
"ra nil",
"ra nil respiratory"
],
[
"nil"
],
[
"passing urine"
],
[
"arrival",
"cf",
"cf patent",
"cf patent flushed",
"flushed",
"patent",
"patent flushed",
"patent flushed pivc",
"patent flushed routine",
"pivc",
"pivc cf",
"pivc cf patent"
],
[],
[]
]
Patient care requirements
[]

```

```

Staff competencies
[
  "",
  "Advanced life Support",
  "Blood Sampling and Cannulation",
  "Arterial Sheath Removal",
  "Inotropic Management"
]
Staff has appropriate competencies for Patient = true

```

The TF-IDF program does return a true result and completes the assignment of the patient to a staff member, while this result does provide a level of redundancy, key words do exist within the note and the correct staff member should be assigned.

Model parameter adjustments: Variation of parameters such as `min_df` and `max_df` is either not possible in the case of `max_df` or non-impacting in the case of `min_df`. Due to the “reverse” nature of the implementation used for TF-IDF and this projects function the `max_df` function cannot be altered, any number less than 1.0 returns an error, “all terms have been pruned”, indicating that no words exists in less than some percentage of the documents. As only one text document is being considered a words must exist in exactly 100% of the documents and therefore `max_df` must be configured to 100% or 1.0. Altering `min_df` does not impact the results of the key word identification as this parameter adjusts the minimum number of documents a term exists in, therefore, for a single document a match cannot be anything less than 100% of the documents.

The number of key words identified and return for comparison is defined by the parameter setting `max_features`, reducing the max features parameters resulted in varying results of key word identification, ideally this parameter would be kept to a minimum when considering scaling of the project for the future but testing showed an optimal number of `max_features` (40) appears to give the most consistent results for identifying as many key words as possible.

The ngram range parameter is required to be in line with the longest phrase in the key word json files, a trigram. If these parameters do not match, then critical key phrases may be missed.

6.3 Model 2 – Topic Modelling with TF-IDF and NMF

Addition of NMF collects terms identified by the TF-IDF vectorizer and groups results into topics based on how similar words appear in a word vector. The example graphs shown in figure 3 display the top 5 words recorded for each topic produced by the NMF model, when comparing these results with the key phrase list no matches are recorded. An inspection of all key phrase results in a single topic did produce some useable results as shown in the progress note results file seen below.

NMF Parameter adjustments: The `n_components` parameter (number of topics) alters the results of the model by increasing the number of topics, extracted features with in a topic a more closely related, this does not provide any significant improvement or degradation to the overall result of the model. `solver`, `beta_loss` and `top words` parameters were all adjusted in an attempt to improve the results of the model, however, no improvements above what has been shown here were witnessed. As testing and evaluation was continued

it was determined that TF-IDF with NMF was not suitable for this application. The NMF model is to predict a topic based on several bodies of text, this project looks to extract critical and often unique pieces of information from a single body of text.

Nurse Note 1 topic modelling results

An example of topic modelling displaying the top 5 words of each topic (n_component) is shown below. Results grouped in this format provide no key phrase matches.

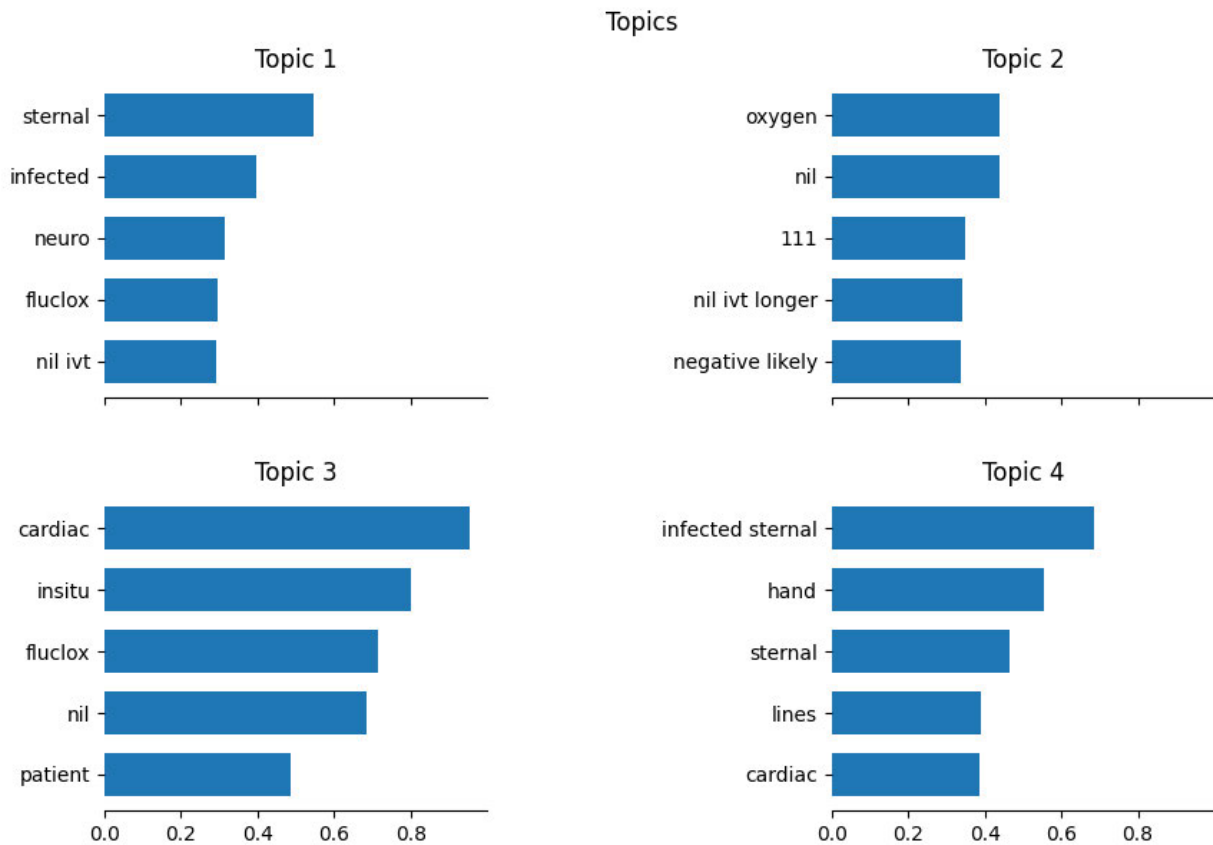


Figure 3: NMF topic modelling results

NMF Nurse Progress Note 2, Staff Member 2 Results,

Key phrase extraction only identifies one competency and patient assignment has not been completed.

Key words found in progress note

[
 "peripheralty warm",
 "prostate issues lines",
 "sheath",
 "pain",
 "radial tr band",
 "peripheralty warm",
 "pt states",

```
"patent flushed",  
"post",  
"pain",  
"omls nad resp",  
"post",  
"radial tr",  
"sheath",  
"nil"
```

```
]
```

Patient care requirements

```
[
```

```
"Arterial Sheath Removal",  
"Arterial Sheath Removal"
```

```
]
```

Staff competencies

```
[
```

```
"",  
"Wound Management",  
"Basic Life Support",  
"Chest Pain Management",  
"STEMI Management"
```

```
]
```

Staff has appropriate competencies for Patient = false

6.4 Model 3 – Multiclass text classification with Keras and TesnorFlow

Results of the multiclass text classification model have been inconclusive. The model runs without error and is capable of a prediction, however results converge to an accuracy of 1 within 1 – 2 epochs, indicating that the model is significantly over fitting. An example model summary can be seen in figure 4 and shows the current configuration of the model, including number of inputs, outputs, layers and number of parameters. The model example has the following hyperparameters configured,

- LSTM_Layer_1 = 64
- Batch size = 16
- Epochs = 8
- Validation split = 0.2

Model: "model"		
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 250)]	0
embedding (Embedding)	(None, 250, 300)	3156600
lstm (LSTM)	(None, 512)	1665024
dense (Dense)	(None, 6)	3078
Total params: 4,824,702		
Trainable params: 1,668,102		
Non-trainable params: 3,156,600		

Figure 4: Multiclass model summary

Model scores can be seen below in figure 5 and show the model converging after only 2 epochs, there may be several reasons for the overfitting of the model including,

- poor implementation of code
- lack of variance in test data caused by hypothetical generation based on the 10 examples
- word embeddings file not suitable for application.

```
Epoch 1/8
21/21 [=====] - 34s 2s/step - loss: 0.6408 - acc: 0.9911 - val_loss: 0.5135 -
val_acc: 1.0000
Epoch 2/8
21/21 [=====] - 36s 2s/step - loss: 0.5732 - acc: 0.9970 - val_loss: 0.5159 -
val_acc: 1.0000
Epoch 3/8
21/21 [=====] - 35s 2s/step - loss: 0.5726 - acc: 0.9970 - val_loss: 0.5251 -
val_acc: 1.0000
Epoch 4/8
21/21 [=====] - 41s 2s/step - loss: 0.5764 - acc: 0.9970 - val_loss: 0.5210 -
val_acc: 1.0000
Epoch 5/8
21/21 [=====] - 41s 2s/step - loss: 0.5749 - acc: 0.9970 - val_loss: 0.5115 -
val_acc: 1.0000
Epoch 6/8
21/21 [=====] - 34s 2s/step - loss: 0.5707 - acc: 0.9970 - val_loss: 0.5163 -
val_acc: 1.0000
Epoch 7/8
21/21 [=====] - 33s 2s/step - loss: 0.5715 - acc: 0.9970 - val_loss: 0.5129 -
val_acc: 1.0000
Epoch 8/8
21/21 [=====] - 35s 2s/step - loss: 0.5706 - acc: 0.9970 - val_loss: 0.5185 -
val_acc: 1.0000
4/4 [=====] - 3s 655ms/step - loss: 0.5262 - acc: 1.0000
Test Score: 0.5261887907981873
Test Accuracy: 1.0
```

Figure 5: Multiclass Model outcomes

Model Testing: The model is loaded by a separate python script and tested against some “unseen data” results of which can be seen below. The results here are depicted in class relevance, that is, the model returns a number between 0 and 1 for each competency (column) relating to how likely it predicts that the progress note belongs to a competency. Little to no variance exists between unseen data tested and this again is likely due to concerns listed above. An example result can be seen from testing nurse note 7 in figure 6.

Model: "model"		
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 250)]	0
embedding (Embedding)	(None, 250, 300)	3156600
lstm (LSTM)	(None, 512)	1665024
dense (Dense)	(None, 6)	3078
Total params: 4,824,702		
Trainable params: 1,668,102		
Non-trainable params: 3,156,600		
<pre>[[0.46198413 0.06744635 0.16702463 0.17032765 0.06892826 0.06428905] [0.46198413 0.06744635 0.16702463 0.17032765 0.06892826 0.06428905] [0.46198404 0.06744635 0.16702461 0.17032763 0.06892826 0.06428903] [0.46198407 0.06744636 0.16702461 0.17032766 0.06892826 0.06428904] [0.46198407 0.06744636 0.16702461 0.17032765 0.06892826 0.06428904] [0.46198404 0.06744635 0.16702463 0.17032765 0.06892826 0.06428905]]</pre>		

Figure 6: Multiclass model prediction results

Each column represents a competency from the .csv file loaded during the model training, from left to right column representation is "12 Lead ECG", "Wound Management", "Arterial Sheath Removal", "Basic Life Support", "Inotropic Management", "TR Band Management".

Some results do appear appropriate, from the training data the 12 Lead ECG competency was the most represented and is reflected in the results in column 1.

Unfortunately, time constraints have prevented the over fitting and poor results of the multiclass model from being rectified, further work in this space is needed.

6.5 Model 4 – Named Entity Recognition with Spacy

The Spacy NER model returns accurate key word matches for each nurse progress note and saves results in the same format as shown in early model results. Results of progress note 2 for staff 1 and 2 and varying staff with varying competencies can be seen below. The Spacy NER model is accurately able to pass identified named entities as competency requirements to the allocation program, staff.py, which returns staff to patient allocation.

Nurse Progress Note 2, Staff Member 3

The spacy model identifies all key phrases for progress note 2 and correctly allocates the patient to staff member 3. For nurse progress note 2 the SpaCy model returns the same competency requirement twice, this is caused by multiple key terms of the one competency being identified and therefore, the program is reporting the competency twice, while this is not desirable it was not considered to be impacting on results for the scope of this project. This error appears in the analysis of several progress notes.

Key words found in progress note

```
[
  "Telemetry",
  "SR",
  "femoral",
  "arterial",
  "TR",
  "band"
]
Patient care requirements
[
  "12 Lead ECG",
  "Arterial Sheath Removal",
  "Arterial Sheath Removal",
  "TR Band Management"
]
Staff competencies
[
  "TR Band Management",
  "Advanced Life Support",
  "12 Lead ECG",
  "PICC Line",
  "Arterial Sheath Removal"
]
Staff has appropriate competencies for Patient = true
```

Nurse Progress Note 9, Staff Member 4

Progress note 9 results show all appropriate key phrases identified and no allocation complete to staff member 4 based on available competencies.

Key words found in progress note

```
[
  "Telemetry",
  "SR",
  "anterior",
  "PCI",
  "TR",
  "band",
  "protocol"
]
```

Patient care requirements

```
[
  "12 Lead ECG",
  "STEMI Management",
  "TR Band Management"
]
```

Staff competencies

```
[
  "Pacing Wire Removal",
  "Epidural and Regional Block Pain",
  "Thrombolysis Management",
  "Basic Life Support",
  "Advanced Life Support"
]
```

Staff has appropriate competencies for Patient = false

Nurse Progress Note 6, Staff Member 2 Results,

Results of progress note 6 returned a match on one competency, an improvement on other models, however, one competency extraction has been missed. Patient allocation has been successfully completed for staff member 2 based on current results.

Key words found in progress note

```
[
  "Telemetry",
  "SR"
]
```

Patient care requirements

```
[
  "12 Lead ECG"
]
```

Staff competencies

```
[
  "",
  "Advanced life Support",
  "Blood Sampling and Cannulation",
  "Arterial Sheath Removal",
  "12 Lead ECG"
]
```

Staff has appropriate competencies for Patient = true

The accuracy and reliability of the NER model depends heavily on the creation of the model and the named entity key_word_doc used. If key words are not entered during the training process, these will not be labelled with the entity name and not identified later during validation.

6.6 Model Accuracy

The following sets of tables show the accuracy for each of the models tested. Total key words extracted by the model in some instances exceed the expected number of matches, this relates to multiple extraction of the same key word. Key word accuracy has been calculate based on all key words (correct and incorrect) extracted by the model divided by the number of expected keywords. This indicates how many terms it takes the model to find a key term, and this is recorded under <model name> Key word accuracy.

Model Competency accuracy is determined based on the number of patient care requirements identified (shown in above examples) by the staff.py allocation function divided by the expected number of competencies per progress note. It is important to note that the program does return duplicate patient care requirements as outlined above and these duplicates have been removed for competency accuracy calculations.

Table 3: shows the accuracy results of the TF-IDF model based on the 10 original example notes.

Table 3: TF-IDF results

Progress Note	TF-IDF Key word accuracy	TF-IDF Competency Accuracy
1	0.1818	0.4
2	0.1428	0.3333
3	0	0
4	0.3333	0.5
5	0.375	0.75
6	0	0
7	0.4285	0.6667
8	0.3333	0.3333
9	0.1667	0.33333
10	0.6	1
Average	0.2561	0.4317

Table 4: shows the accuracy results of the TF-IDF +NMF model based on the 10 original example notes.

Table 4: TF-IDF+NMF results

Progress Note	TF-IDF +NMF Key word accuracy	TF-IDF +NMF Competency Accuracy
1	0.0909	0.2
2	0.4285	0.3333
3	0	0
4	0.3333	0
5	0.375	0.5
6	0	0
7	0	0
8	0.1667	0.3333
9	0	0
10	0.4	0.5
Average	0.1794	0.1867

Table 5: shows the accuracy results of the SpaCy NER model based on the 10 original example notes.

Table 5: SpaCy NER results

Progress Note	SpaCy Key word accuracy	SpaCy Competency Accuracy
1	1.182	0.8
2	0.8571	1
3	2	1
4	1.667	1
5	1	1
6	0.5	0.5
7	1.143	1
8	1.333	0.6667
9	1.167	1
10	1.4	1
Average*	1.2248	0.8967

*Average has been shown as a score above 1 due to model returning a match for unigram, bigram and/or trigram for one key word/phrase within the note (multiple matches for one event).

6.7 Chapter Summary

The results of the NLP models have been presented in example output files which were analysed for successful key word matches, configuration adjustments were made to find the optimal performance. Each model outcome has then been analysed for accuracy over key words found and competency matches these have been presented in a series of tables. No results of the multiclass text classification model have been presented in terms of key word matching or competency matching.

Chapter 7. Conclusions and Further Work

7.1 Achievement of Project Objectives:

The following objectives have been addressed with varying success:

Research impacts of implementing the optimal staff-to-patient allocation

Results of the literature review are shown in chapter 2, this explains in detail the nursing staff structure for a shift and outlines the need for appropriate staffing and skill mix for individual patients. The literature also indicates that an automated solution to patient-to-staff allocation would have a positive impact if the requirements of reduced admin time and allocation accuracy are met.

Research and test techniques of Natural Language Processing and Machine Learning

Evidence of this research is shown in chapter 3 where critical techniques of NLP are presented and how these approaches can be applied over machine learning models. Four approaches are identified for testing.

Data Collection

Chapter 4 outlines the acquisition of data and how it was been managed. The original goal of the project was to acquire real world data from an appropriate facility, this approach would have provided the large volumes of data required to accurately train the machine learning models and also provide a true review of the systems capabilities. Due to time constraints and delays in approval by the facility, access to the data was not achieved and an alternate method for volume data was developed. The generated data has provided a certain level of testing of the NLP and ML approaches, however, results for the Multiclass text classification appear non-valid.

Design and implement an automated solution

All four of the nominated approaches have been developed and simulated as shown in chapter 5. Each model can return a list of competencies extract from a single progress note. A Class was developed for hypothetical staff members and competency attributes assigned, these could then be compared to a patient's requirements and patient-to-staff allocation achieved. The project was intended to also consider a staff members experience and education, but this has not been implemented due to time constraints.

Analyse the results

Chapter 6 gives a detailed explanation of the results achieved from each model. Results provide evidence to support the capability that NLP and ML can be used to achieve an automated solution to staff-to-patient allocation process. A summary of each model's success can be seen in the tables 3 - 5. Importantly the result of the SpaCy NER model does indicate that this approach could be successful in a real-world application, with some limiting parameters.

7.2 Further Work

The project has managed to implement some basic NLP techniques for a unique data set and adequately proven that the solution is achievable through to a production level. Although the project has been partly successful some key items for further work remain,

- Testing and analysis with real world data
- Handling of unknown entities or competencies
- Spelling mistakes
- Repeat notes for one patient

The primary candidate for further work is to use and test a large volume of real-world data, as outlined previously this was the intent of the project. Real world data introduces a significantly larger number of variables including spelling, note structure, language, and unseen data. Based on current model designs I do believe that real world data would significantly reduce the accuracy of all models as many of the aforementioned items have not been considered.

Further work with real world data would allow for multiple progress notes from one individual to be reviewed by the system. Multiple notes from a single patient may provide an opportunity to identify trends in a patient's recovery and possibly even predict outcomes to further improve patient recovery. Models such as the TF-IDF and NMF may also increase in accuracy if multiple documents exist for one patient.

Lastly a larger neural network to support feedback and additional inputs could help to solve expected issues such as spelling mistakes or correctly categorise unseen data. It would be expected that real world data would continuously provide unseen data, and this would need to be considered for the solution to remain successful.

References

- Sheikhalishahi, S, Miotto, R, Dudley, JT, Lavelli, A, Rinaldi, F & Osmani, V 2018, 'Natural Language Processing of Clinical Notes on Chronic Diseases: Systematic Review', *JMIR Medical Informatics*, vol. 7, no. 2, viewed 15 May 2021, <<https://medinform.jmir.org/2019/2/e12239/>>.
- Twigg, D & Duffield, C 2009, 'A review of workload measures: a context for a new staffing methodology in Western Australia', *International Journal of Nursing Studies*, viewed 20 May 2021, <<https://pubmed.ncbi.nlm.nih.gov/18789439/>>
- Prashant, N, Deven, J, Omender, S, Rohit, D & Vikas, A 2011, 'Severe sepsis and its impact on outcome in elderly and very elderly patients admitted in intensive care unit', *Journal of Intensive Care Medicine*, viewed 20 May 2021, <<https://pubmed.ncbi.nlm.nih.gov/21436163/>>
- Aiken, L, Clarke, S & Sloane, D 2002, 'Hospital Nurse Staffing and Patient Mortality, Nurse Burnout, and Job Dissatisfaction', *The Journal of American Medical Association*, viewed 20 May 2021, <<https://jamanetwork.com/journals/jama/fullarticle/195438>>
- Saville, C, Griffiths, P, Ball, J & Monks, T 2019, 'How many nurses do we need? A review and discussion of operational research techniques applied to nurse staffing', *International Journal of Nursing Studies*, viewed 20 May 2021, <<https://pubmed.ncbi.nlm.nih.gov/31129446/>>
- Maenhout, B & Vanhoucke, M 2013, 'An integrated nurse staffing and scheduling analysis for longer-term nursing staff allocation problems', *Omega*, viewed 20 May 2021, <<https://ideas.repec.org/a/eee/jomega/v41y2013i2p485-499.html>>
- Flinter, M, Hsu, C, Crompt, D, Ladden, M, Wagner, E 2017, 'Registered nurses in primary care: emerging new roles and contributions to team-based care in high-performing practices', *The Journal of Ambulatory Care Management*, viewed 20 May 2021, <<https://pubmed.ncbi.nlm.nih.gov/28323721/>>
- Peerson, A, Aitken, R, Manias, E, Parker, J, Wong, K 2002, 'Agency nursing in Melbourne, Australia: a telephone survey of hospital and agency managers', *Journal of Advanced Nursing*, viewed 20 May 2021, <<https://pubmed.ncbi.nlm.nih.gov/12437599/>>
- Khanna, S, Good, N & Lind, J 2013, 'Operational efficacy of the midnight census', *British Journal of Healthcare Management*, viewed 20 May 2021, <<https://publications.csiro.au/rpr/pub?pid=csiro:EP131306>>
- Silver, P & Sweberg, T 2013, 'Patient census at midnight does not accurately reflect peak bed utilization in an ICU', *Critical Care Medicine*, viewed 20 May 2021, https://journals.lww.com/ccmjournal/abstract/2013/12001/3__patient_census_at_midnight_does_not_accurately.7.aspx
- Cox, H, James, J & Hunt, J 2006, 'The experiences of trained nurses caring for critically ill patients within a general ward setting', *Intensive and Critical Care Nursing*, viewed 20 May 2021, <<https://pubmed.ncbi.nlm.nih.gov/16621564/>>
- Pantazopoulos, I, Tsoni, A, Kouskouni, E, Papadimitriou, L, Johnson, E & Xanthos, T 2012, 'Factors influencing nurses' decisions to activate medical emergency teams', *Journal of Clinical Nursing*, viewed 20 May 2021, <<https://pubmed.ncbi.nlm.nih.gov/22889450/>>
- McDonnell, A, Tod, A, Bray, K, Bainbridge, D, Adsetts, D & Walters, S 2013, 'A before and after study assessing the impact of a new model for recognizing and responding to early signs of deterioration in an acute hospital',

Journal of Advanced Nursing, viewed 20 May 2021, <<https://onlinelibrary.wiley.com/doi/10.1111/j.1365-2648.2012.05986.x>>

Chua, W, Mackey, S, Ng, E & Liaw, S 2013, 'Front line nurses' experiences with deteriorating ward patients: a qualitative study', *International Nursing Review*, viewed 20 May 2021, <<https://pubmed.ncbi.nlm.nih.gov/24131252/>>

Hart, P, Spiva, L, Baio, P, Huff, B, Whitfield, D, Law, T, Wells, T & Mendoza, I 2014, 'Medical-surgical nurses' perceived self-confidence and leadership abilities as first responders in acute patient deterioration events', *Journal of Clinical Nursing*, viewed 20 May 2021, <<https://pubmed.ncbi.nlm.nih.gov/24393472/>>

Liaw, S, Scherpbier, A, Klainin-Yobas, P & Rethans, J 2011, 'A review of educational strategies to improve nurses' roles in recognizing and responding to deteriorating patients', *International Nursing Review*, viewed 20 May 2021, <<https://pubmed.ncbi.nlm.nih.gov/21848774/>>

Duffield, C, Roche, M, Blay, N, Thomas, D & Stasa, H 2011, 'The consequences of executive turnover', *Journal of Research in Nursing*, viewed 20 May 2021, <<https://opus.lib.uts.edu.au/bitstream/10453/18690/1/2011003536.pdf>>

Hayes, L, O'Brien-Pallas, L, Duffield, C, Shamian, J, Buchan, J, Hughes, F & North, N 2012, 'Nurse turnover: a literature review—an update', *International Journal of Nursing Studies*, viewed 20 May 2021, <<https://pubmed.ncbi.nlm.nih.gov/22019402/>>

O'Brien-Pallas, L, Murphy, G, Shamian, J, Li, X & Hayes, L 2010, 'Impact and determinants of nurse turnover: a pan-Canadian study', *Journal of Nursing Management*, viewed 20 May 2021, <<https://pubmed.ncbi.nlm.nih.gov/21073578/>>

Eley, R, Buikstra, E, Plank, A, Hegney, D & Parker, V 2007, 'Tenure, mobility and retention of nurses in Queensland, Australia: 2001 and 2004', *Journal of Nursing Management*, viewed 20 May 2021, <<https://pubmed.ncbi.nlm.nih.gov/17359428/>>

Cameron, I, Gillespie, D, Robertson, M, Murray, G, Hill, K, Cumming, R 2012, 'Interventions for preventing falls in older people in care facilities and hospitals', *Cochrane Database of Systematic Reviews*, viewed 20 May 2021, <<https://pubmed.ncbi.nlm.nih.gov/23235623/>>

Pabico, C, Graystone, R 2018, 'Comparing pathway to excellence and magnet recognition programs', *American Nurse Today*, viewed 20 May 2021, <<https://www.myamericannurse.com/comparing-pathway-excellence-magnet-recognition-programs/>>

Dunton, N, Gajewski, B, Klaus, S 2007, 'The relationship of nursing workforce characteristics to patient outcomes', *Online Journal of Issues in Nursing*, viewed 20 May 2021, <https://www.researchgate.net/publication/26571298_The_Relationship_of_Nursing_Workforce_Characteristics_to_Patient_Outcomes>

Cho, S, Ketefian, S, Barkauskas, V 2003, 'The effects of nurse staffing on adverse events, morbidity, mortality, and medical costs' *Nursing Research*, viewed 20 May 2021, <https://pubmed.ncbi.nlm.nih.gov/12657982/>

Queensland Health 2016, 'Nurse-to-patient ratios- Questions and answers', Office of the Chief Nursing and Midwifery Officer, viewed 20 May 2021, <https://www.health.qld.gov.au/__data/assets/pdf_file/0027/357453/ratiosqa.pdf>

Diego Lopez Yse 2019, 'Your Guide to Natural Language Processing (NLP)', *Medium, Towards Data Science*, viewed 28 April 2021, <<https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1>>.

Liddy, E 2001, *Natural Language Processing Recommended Citation*, *Center for Natural Language Processing School of Information Studies (iSchool)*, p.2001, viewed 22 Aug. <https://surface.syr.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1019&context=cnlp>

Python Software Foundation 2019, *Welcome to Python.org*, Python.org, Python.org.

spaCy · Industrial-strength Natural Language Processing in Python 2015, spaCy, viewed 1 March 2021, <<https://spacy.io/>>.

TensorFlow 2019, *TensorFlow*, viewed 3 August 2021, <<https://tensorflow.org>>

Team, K 2019, *Keras documentation: Introduction to Keras for Engineers*, viewed 3 August 2021 <<https://keras.io/>>

scikit-learn developers 2019, *User guide: contents — scikit-learn 0.22.1 documentation*, viewed 27 July 2021 <<https://Scikit-learn.org/>>

Appendix A

Project specifications

ENG4111/4112 Research Project

Project Specification

For: Daniel Price

Title: Natural Language Processing AI to support Nurse-Patient allocation in acute care

Major: Computer Systems

Supervisors: Mark Phythian

Enrollment: ENG4111 – EXT S1, 2021

ENG4112 – EXT S2, 2021

Project Aim: Effectively implement the optimal model of nursing in acute care using AI and Natural Language processing. That is, automatically allocate nursing staff based on staff experience, skills and competencies to patients based on conditions as documented in the patient notes.

Program: Version 1, 17th March 2021

1. Conduct initial background research on Nursing care models in Australian acute care. Conduct initial background research into AI/NLG and compare algorithms, programming languages, training methods and implementation difficulty.
2. Select a Hospital and ward from which to review current methods and practices.
3. Assess hardware requirements for necessary capability and costs.
4. Select hardware and a suitable software development environment.
5. Identify key measurable factors for patient and staff satisfaction, outline current baseline performances.
6. Identify key sentences/words in notes that can define patient condition, define a set of staff competencies and experience.
7. Develop AI algorithm based on research and project goals.
8. Begin training and testing of AI.
9. Complete final training of AI and begin simulation period.
10. Evaluate accuracy and performance of AI to baseline hospital procedures.
11. Complete Dissertation and project review.

If time and resource permit:

12. Deploy AI in Hospital to simulate live environment and collect real time data.
13. Review AI algorithms and improve based on real time results.

Appendix B

Example Project Notes

Note 1:

TOC of patient at 2130 hours.

Dx: infected sternal wound and infected sternal wire.

A-

patent, own

-Resp

breathing spontaneously .

ongoing requirement of supplemental oxygen - 4L/min

to maintain saturations 94%

denies SOB

nil cough

-Cardiac

warm to touch

afebrile ATOR

Not on cardiac Monitoring.

borderline hypotension as per iEMR,

asymptomatic of same

-Neuro

alert and orientated

Mobilising with supervision

-Lines

R) hand and L) hand PIVC insitu

sternal vac dressing insitu - intact, pump attached and working,

Minimal exudate in chamber.

-Gastro

tolerating diet and fluids

nil IVT - as no longer fasting

Regular IVABs

-Lines

BGL at 1800 11.1,

to be checked in mane - on OHAs.

Plan:

For 2nd Daily BCs until negative

Likely 6/52 course flucloxacillin - will need PICC line for same

For further washout in 2-3 days

Flucloxacillin frequency increased to q4H

Note 2:

-Neuro

GCS 15. Alert and orientated.

Denies pain.

-Cardiac

Telemetry monitoring in SR.

BP stable.

Peripherality warm and well perfused.

R) femoral arterial sheath removed at 18:52hrs. ;

Femstop applied post due to small haematoma. Developed as patient non compliant with laying still.

R) radial TR band now Omls. NAD.

-Resp

SpO2 98% RA,

Nil respiratory distress

-Gastro

olerating fluids post sheath rm eov al

BNO

-Pain

Nil C/o nausea.

-Renal

Passing urine in bottle Multiple times this shift.

Pt states has prostate issues,
 -Lines
 PVC in R) CF patent and flushed.

Note 3:

Neuro: Pt alert and orientated - GCS 15

Pt slightly agitated this shift due to not being able to smoke. Pt educated on the Contraindications of smoking. Pt is happy to stay on the ward today.

Requested RMO to chart a second nicotine Patch- administered at the pt's request. Pt otherwise calm and pleasant this shift.

Observations: EWS 0

Monitoring on telemetry - SR

Cardiac: R) radial site dry and intact.

Radial and brachial pulses palpable. Capillary refill less than 2 seconds, Peripheries warm to touch.

Pt states he has chronic intermittent tingling and pain to both arms. Pt states this is from his previous work as a panel beater.

Pt states he has undergone many tests and is on the waiting list for surgery to treat this.

R) femoral site dry and intact. R) Popliteal and dorsalis pedis pulses palpable. Warm peripheries

Capillary refill less than 2 seconds. All neurovascular observations documented in iEMR,

Pt states he has been suffering from restless legs since his last MI in 2020. Pt observed to have restless legs when resting in bed.

Observed to subside when pt is engaged in conversation. TL notified.

Pt has been seen by psychology team for his anxiety and depression,

Pain: Nil c/o pain this shift.

Gastro: Pt tolerates diet and fluids well.

Elimination: Pt PUIT. BO this AM.

Lines: PIVC R) cubital fossa insitu, patent, ni phlebitis.

ADLS/ mobility: Pt independent

Note 4:

Neuro

GCS 14. Orientated to person but not to time or place.

Requiring 1:1 special for impulsiveness and agitation.

Cardiac

Telemetry monitoring in SR. Occasional runs of NSVT.

Asymptomatic with this.

Noted mag level 0.7. Given 10mmol IV Mag.

Noted K level 3. Given total of 20mmol IV K.

BP remains stable throughout.

Peripherally warm and well perfused.

Respiratory

SpO2 95% 2L O2.

SpO2 drops to 85% on RA.

Gastro

Tolerating diet and fluids.

BNO 2 days. Apperients given. Awaiting effect.

Renal

Passing urine in the bottle with assistance.

Lines/Skin

Skin tear on L) arm present on admission. Redressed this shift with Mepilex.

PIVC in R) arm patent and flushed.

Social

Nil visitors this shift. Phone enquiry by daughter. Updated on plan of care.

Note 5:

Received care of patient at 10:15hrs from ICU.

Neuro

GCS 15. Alert and orientated.

Sedation score of 1. Drowsy but able to keep eyes open during conversation.

Fentanyl PCA insitu. 10mcg bolus delivered. Pain score 2/10. Encouraging use of PCA.

PCA line at 6cm. Dressing D+I.

Cardiac

Telemetry monitoring in PR rate of 70.

A + V pacing wires attached to pacing box.

Rate 70. Output 10. Sensitivity 1.0.

BP remains stable.

Peripherally warm and well perfused.

Resp

SpO2 95% 2L O2.

Chest auscultated. Diminished sounds in the bases bilaterally.

Compliant with spirometry.

2:1 ICC insitu. On -20mmHg suction. Swing and bubble on cough.

Draining 40mls 2nd hourly.

Gastro

Tolerating diet and fluids.

BNO 2 days. Given aperients. Awaiting effect.

C/o nausea. Given antiemetics with good effect.

Renal

IDC insitu. Draining 40mls/hr of straw coloured urine.

Wounds

Sternal wound D+I. Comfeel dressing insitu.

R) Radial wound D+I. Comfeel dressing insitu.

Drain dressing remains D+I. Alevyn dressing insitu.

Lines

4x lumen CVL insitu. All lumens patent and flushed.

Dx remains D+I.

PIVC in L) CF patent and flushed.

Social

Wife in attendance this shift. Updated on plan of care.

Note 6:

Received care of patient at 15:30hrs from Lismore Base Hospital.

Thrombolysed at 13:11hrs in LBH.

Neuro

GCS 15. Alert and orientated.

2/24 neurovascular obs attended. PEARL.

Cardiac

ECG attended on arrival to ward.

ST elevation still present but resolving in V1-4.

Noted TWI in inferior leads.

Telemetry monitoring in SR.

Nil arrhythmias noted ATOR.

BP remains stable.

Nil c/o chest pain.

Resp

SpO₂ 95% RA.

Nil respiratory distress.

Gastro

Tolerating diet and fluids.

BSL 20.2. Given STAT Insulin as per CCU Reg.

Ketones 1. Will continue to monitor.

BNO.

Nil c/o nausea.

Renal

Passing urine in the bottle independently.

Lines

PIVC R) CF patent and flushed.

PIVC L) CF patent and flushed.

Routine bloods taken on arrival.

Note 7:

Received care of patient at 12:00hrs from ICU.

D1 post CABG.

Neuro

GCS 14. Drowsy but confused.

Fentanyl PCA insitu. 20mcg bolus. Patient using frequently.

Sedation score 2.

Pt cannot keep eyes open for more than 5 seconds.

PCA button taken away from patient and awaiting treating team R/V for potential decreased dose and Naloxone use.

Cardiac

Telemetry monitoring in PR rate of 70bpm.

Patient connected to pacing box. VVI. Rate 70, Output 10, sensitivity 0.5.

BP remains stable.

Neurovascular obs NAD.

Resp

SpO₂ 94% 3L O₂ via NP.

Need encouragement with spirometry.

2:1 ICC insitu. -20mmHg suction. Swing and BOC.

Draining 20-40mls/hr.

Gastro

Tolerating small amounts of diet and fluids.

BNO.

Nil c/o nausea.

Renal

IDC insitu. Passing 20-40mls/hr.

Lines

4x lumen CVL insitu. All lumens patent and flushing.

PIVC in R) hand patent and flushed. D2.

Note 8:

Received care of patient at 13:45hrs. IHT from LBH.

Noted pt thrombolysed at LBH @ 11:02hrs.

Neuro

GCS 15. Alert and orientated.

Neuro obs attended 1/24 as per lysis protocol.

Neuro obs NAD.

Cardiac

Telemetry monitoring in SR.

ECG attended on arrival and hourly as per STEMI protocol.

STE still present in inferior leads however is improving since admission.

BP remains stable.

Nil c/o chest pain.

Neurovascular obs attended 1/24 as per lysis protocol. NAD.

Resp

SpO2 95% RA.

Nil respiratory distress.

Lung fields clear.

Gastro

Tolerating diet and fluids.

BSL WIL.

BNO.

Renal

Passed urine in the bottle on arrival to CCU.

Lines

PIVC L) CF patent and flushed. D0.

PIVC R) CF patent and flushed. D0.

Note 9:

Received care of patient at 10:45hrs from Cardiac Catheter Lab post Percutaneous Coronary Intervention to Left Anterior Descending artery.

Safety checks attended.

Neuro

GCS 15. Alert and orientated.

Nil c/o pain.

Pt teary upon arrival to CCU. Reassurance and questions answered.

Cardiac

Telemetry monitoring in SR with frequent PVCs. Asymptomatic with this.

BP remains stable.

Noted STE remains in anterior leads however improved since PCI.

Nil c/o chest pain.

Resp

SpO2 94% RA.

Noted patient lifelong smoker.

Lung fields crackles. Given IV Lasix as per CCU Reg.

Gastro

Tolerating diet and fluids.

BSL elevated. Given Novorapid as per sliding scale.

Renal

IDC inserted. Passing 30-50mls/hr.

Lines

PIVC L) CF patent and flushed.

TR band insitu. Air removed as per protocol. Now 0mls.

Neurovascular obs remain NAD.

Note 10:

Received care of patient at 13:15hrs.

Neuro

GCS 15. Alert and orientated.

Nil c/o pain.

Cardiac

Telemetry monitoring in complete heart block and second degree heart block rate of 40bpm.

Isoprenaline continues at 60mcg/hr.

BP remains elevated at 170systolic.

Aiming for HR above 40bpm. Same achieved.

Aiming for MAP above 60. Same achieved.

Patient consented for PPM as per treating team.

Resp

SpO2 95% RA.

Nil respiratory distress.

Lung fields clear.

Gastro

Patient remains NBM in preparation for insertion of PPM.

BNO.

Nil c/o nausea.

Renal

IDC insitu. Passing 20-40mls/hr.

Lines

PIVC L) hand patent and flushed. D2.

PIVC R) CF patent and running isoprenaline. D2.

Appendix C

Staff Competency Files

12lead.json

```
[
  "12 Lead ECG",
  "telemetry",
  "Telemetry",
  "sinus rhythm",
  "SR"
]
```

arterialSheath.json

```
[
  "Arterial Sheath Removal",
  "arterial",
  "Arterial",
  "sheath",
  "arterial sheath",
  "femoral",
  "femoral site"
]
```

basicLife.json

```
[
  "Basic Life Support",
  "oxygen requirement",
  "increasing oxygen",
  "hypotension",
  "O2",
  "oxygen",
  "airways"
]
```

bloodGlucose.json

```
[
  "Blood Glucose Monitoring",
  "BGL",
  "bgl",
  "blood glucose",
  "glucose level",
  "blood sugar",
  "carbohydrate",
  "cellulose",
  "lactose",
  "starch",
  "monosaccharide",
  "dextrose",
  "disaccharide",
  "glycogen",
  "polysaccharide",
  "complex carbohydrate",
  "simple carbohydrate",
  "galactose",
  "fructose",
  "sucrose",
  "xylose"
]
```

bloodSampling.json

```
[
  "Blood Sampling and Cannulation",
  "blood cultures",
  "cannulation"
]
```

Epidural and Regional Block.json

```
[
  "Epidural and Regional Block Pain Management",
  "Fentanyl PCA",
  "PCA",
  "fentanyl",
  "naloxone"
]
```

Epidural Pacing.json

```
[
  "Epicardial Pacing",
  "pacing box",
  "VVI",
  "PR",
  "pacing",
  "pacing wire"
]
```

Inotropic Management.json

```
[
  "Inotropic Management",
  "complete heart block",
  "second degree heart block",
  "isoprenaline",
  "heart block"
]
```

piccLine.json

```
[
  "PICC Line Management",
  "PICC line",
  "PICC"
]
```

STEMI Management.json

```
[
  "STEMI Management",
  "cardiac catheter lab",
  "catheter lab",
  "cardiac catheter",
  "percutaneous coronary intervention",
  "left anterior descending",
  "anterior descending",
  "PCI"
]
```

Thrombolysis Management.json

```
[
  "Thrombolysis Management",
  "lysis",
  "thrombolysed",
]
```

```
"lysis protocol",  
"Thrombolysed",  
"Thrombolysis",  
"thrombolysis"  
]
```

TRband.json

```
[  
  "TR Band Management",  
  "TR band",  
  "TR",  
  "haemostasis",  
  "radial artery",  
  "radial compression"  
]
```

woundManagment.json

```
[  
  "Wound Management",  
  "Wound",  
  "wound",  
  "Injury",  
  "injury",  
  "cut",  
  "slash",  
  "lesion",  
  "laceration",  
  "infected",  
  "dressing",  
  "damage",  
  "bleeding",  
  "lacerate",  
  "gash",  
  "exudate",  
  "vac dressing",  
  "skin tear",  
  "tear",  
  "comfeel",  
  "alevyn",  
  "alevyn dressing"  
]
```

Appendix D

Key Words List

key_doc.json

```
[
  "hypotension",
  "infected",
  "Infected",
  "Arterial",
  "arterial",
  "Telemetry",
  "telemetry",
  "sinus",
  "rythm",
  "Sinus",
  "Rythm",
  "SR",
  "TR",
  "band",
  "site",
  "femoral",
  "Femoral",
  "Wound",
  "wound",
  "vac",
  "Vac",
  "dressing",
  "exudate",
  "Oxygen",
  "oxygen",
  "requirement",
  "increasing",
  "Hypotension",
  "BGL",
  "Blood",
  "blood",
  "Glucose",
  "glucose",
  "level",
  "sugar",
  "cultures",
  "PICC",
  "line",
  "Fentanyl PCA",
  "PCA",
  "fentanyl",
  "naloxone",
  "pacing box",
  "VVI",
  "PR",
  "complete heart block",
  "second degree heart block",
  "isoprenaline",
  "heart block",
  "cardiac catheter lab",
  "catheter lab",
  "cardiac catheter",
  "percutaneous coronary intervention",
  "left anterior descending",

```

"anterior descending",
"PCI",
"lysis",
"thrombolysed",
"lysis protocol",
"skin tear",
"tear",
"comfeel",
"alevyn",
"alevyn dressing",
"Injury",
"injury",
"cut",
"slash",
"lesion",
"laceration",
"damage",
"bleeding",
"lacerate",
"gash",
"cannulation"

]

Appendix E

GCUH Standard Student Research Agreement

NOTE: This Agreement is for non-interventional research being undertaken by a Student under the supervision of the University. The University and Student/s wish to request information or assistance (or both) from a Hospital and Health Service which may include access to patients and/or patient information held by that Hospital and Health Service. The Hospital and Health Service agree to provide the University and Student/s with information and assistance in accordance with this Agreement. If the University or Student is not obtaining patient consent prior to accessing patient data (i.e. asking the HREC to waive the requirement for consent) a PHA application will be required.

The Agreement is not to be used where:

1. the Hospital and Health Service is actively involved in collaborating on the research and will be creating intellectual property (in this case, a collaboration agreement should be used);
2. the research involves clinical trials or treating patients; or,
3. the Student only requires access to patients for interviews only or use of SCHHS equipment only (in this case, a Facility Access Agreement should be used).

STUDENT AND UNIVERSITY RESEARCH AGREEMENT

This Agreement is made on the date the last party signs it.

BETWEEN: **GOLD COAST HOSPITAL AND HEALTH SERVICE** ABN 82 616 992 416, 1 Hospital Boulevard Southport, QLD, 4215 Australia (**the HHS**)

AND: **UNIVERSITY OF SOUTHERN QUEENSLAND** ABN 40 234 732 081 of West Street, Darling Heights QLD 4350

AND ACKNOWLEDGED AS A THIRD PARTY BY

Daniel Price of 87 Honeyeater Drive, Burleigh Waters, Gold Coast, QLD 4220 (**the Student**).

BACKGROUND

- A The Student is undertaking the Project under the supervision and with the support of the University in accordance with the Proposal attached in **Schedule 2**.
- B The Student is enrolled in a formal program of study as specified in **Schedule 1**.
- C The University acknowledges the Project involves non-interventional research to be undertaken by the Student under the supervision and with the support of the University.
- D The University and Student require the information and/or assistance of the HHS for the purpose of the Project, as specified in **Schedule 1**.
- E The HHS agrees to provide the HHS Assistance specified in **Schedule 1** for the purpose of the Project subject to the terms and conditions of this agreement.

DEFINITIONS

Agreement means this document and all schedules and annexure to it.

Background Intellectual Property means information, techniques, know-how, software and materials (regardless of the form or medium in which they are disclosed or stored) that are provided by one party to the other for use in the Project (whether before or after the date of this Agreement), and all Intellectual Property in them.

Clinical Subject means any human subject involved in the Project or whose data will be handled in the course of the Project, including in the course of carrying out this Agreement.

Clinical Subject Data means data or other information collected from a Clinical Subject or created in the course of clinical treatment or observation about a Clinical Subject in the course of the Project and includes all medical records in relation to the Clinical Subject used in the course of the Project but does not include Clinical Subject Materials.

Code means the Australian Code for the Responsible Conduct of Research issued by the National Health and Medical Research Council.

Commencement Date means the date on which this agreement is signed by both Parties. If the Parties sign the Agreement on different dates, then the commencement date is the later of those dates.

Completion Date means the date specified in **Schedule 1**.

Commercialisation Lead mean:

- (a) the entity named to carry out commercialisation in **Schedule 1**; or,
- (b) if no entity is set out in **Schedule 1** as the Commercialisation Lead, then the Party appointed as owner of New Intellectual Property under clause 5.3; or,
- (c) if there is no Commercialisation Lead in **Schedule 1** and no entity is set out as owner of New Intellectual Property under clause 5.3, then clause 5.5(d) shall be of no effect.

Confidential Information means any information passed by one Party to the other Party that is, or ought to reasonably be known to be, secret but does not include information that is:

- (a) in the public domain;
- (b) the Parties agree is not confidential;
- (c) independently discovered or received by the other Party without reference to the information disclosed under this Agreement; or,
- (d) provided to the University in accordance with consent of a Clinical Subject.

Ethics Committee means the human research ethics committee or other appropriate ethics committee specified in the Schedule.

Ethics Approval means the documents, including any NEAF, which is submitted, anticipated and approved by the Ethics Committee.

Facility means the hospital or health facility within the HHS specified in **Schedule 1**.

HHS means the Hospital and Health Service identified on page 1 and in **Schedule 1**.

HHS Assistance means the information and/or assistance to be provided by the Facility

HHS Contact means the person specified in **Schedule 1**.

Intellectual Property means all intellectual property rights, including but not limited to:

- (a) trade and service marks (including goodwill in those marks), patents, inventions, discoveries, copyright, rights in circuit layouts, designs, domain names, registrable plant varieties or processes;
- (b) any application or right to apply for registration of any rights referred to in paragraph (a); and
- (c) all rights of a similar nature to any of the rights in paragraph (a) and (b) which may subsist anywhere in the world (including Australia), whether or not such rights are registered or capable of being registered.

New Intellectual Property means Intellectual Property created during the course of or arising from the Project.

Party means the HHS or the University or both as the context dictates.

Personal Information is information or an opinion, including information or an opinion forming part of a database, whether true or not, about an individual whose identity is apparent, or can reasonably be ascertained, from the information or opinion.

Project means the project as described **Schedule 1** and further detailed in **Schedule 2**.

Protocol means a protocol, if necessary, for carrying out the project as approved by the relevant Ethics Committee.

Publish/Publication means to publish by way of a paper, article, manuscript, report, poster, internet posting, presentation, abstract, outline, video, instruction material or other disclosure, in printed, electronic, oral or other form.

Relevant Privacy Laws means the *Information Privacy Act 2009* (Qld), *Hospital and Health Boards Act 2011* (Qld) and any other legislation (including delegated and subordinate legislation such as regulations), code or guideline which applies in the jurisdiction where the Project is to be conducted and which relates to the protection of Personal Information.

Schedule 1 – Agreement Details

Name of Facility operated by the HHS providing the HHS Assistance	Gold Coast Hospital and Health Service
Student Names and Programs of Study (if applicable)	Daniel Price Engineering (Undergraduate)
Investigators names and contact details	Daniel Price - [REDACTED] Mark Phythian - Mark.Phythian@usq.edu.au (Supervisor)
Description of Project (further details included in Ethics Approval in Schedule 2)	Natural Language Processing AI to support Nurse-to-Patient allocation in acute care
New Intellectual Property owner	University of Southern Queensland
Commercialisation Lead	Not applicable
HHS Assistance	Provision of de-identified patient notes.
Name of HHS Contact	Vanessa Druett Research Governance Lead GCHResearch@health.qld.gov.au Ph: 07 5687 3880
Completion Date	Upon the end of ethics approval
Ethics Committee	GCHHS HREC (ERM Reference 77280)
Acknowledgement of HHS assistance	The University and the Students will acknowledge the assistance of the HHS in all publications arising from the Project in the following format: <i>We gratefully acknowledge the support provided by Gold Coast Hospital and Health Service in the conduct of this research.</i>
Reimbursement of Expenses	Not Applicable.
Additional Conditions	Not Applicable.
Notices	HHS: Name: Dr Greta Ridley Position: Director of Research Telephone Number: 07 5687 3880 Postal Address: Office for Research Governance and Development Level 2, Pathology and Education Building

	<p>1 Hospital Boulevard, Southport QLD 4215 Email: GCHResearch@health.qld.gov.au</p> <p>University: Name: Mark Pythian, University of Southern Queensland Position: Program Director – Undergrad Engineering Telephone Number: 07 4631 2542 Postal Address: USQ, Office of Research (Administration), Toowoomba Campus, QLD 4350</p> <p>Student: Name: Daniel Price Position: Engineering Student – under grad Telephone [REDACTED] [REDACTED] [REDACTED] [REDACTED]</p>
--	--

Appendix F

USQ Endorsement Letter



1 July 2021

Mark Phythian
Program Director – Undergrad Engineering
Faculty of Health, Engineering and Sciences
University of Southern Queensland

P: (07) 4631 2542
E: phythian@usq.edu.au

To the Ethics Committee – Gold Coast University Hospital

Re: Endorsement of research project by USQ student

Daniel Price - [REDACTED]

In my capacity as Daniel Price's research project supervisor, I provide my full support for his project titled "Natural language processing AI to support nurse-to-patient allocation in acute care".

I endorse his application for ethics approval to the GCUH Ethics Committee.

Daniel has also made an application to the USQ Ethics Committee with me listed as co-investigator. Their initial feedback suggests that the USQ Ethics Committee would automatically approve this project once the GCUH Ethics Committee has provided approval.

Please contact me (as above) if you require further information.

Regards

[REDACTED]
Mark Phythian
Program Director – Undergrad Engineering

Appendix G

Python Scripts

G.1 *Staff.py* The Staff Allocation script

```

import os
import json

def load_data(file):
    with open(file, "r", encoding = "utf-8") as f:
        data = f.read()
    return(data)

class Nurse:
    def __init__(self, ID, Education, experience1, experience2, comp1, comp2,comp3, comp4, comp5):
        self.ID = ID
        self.Education = Education
        self.experience1 = experience1
        self.experience2 = experience2
        self.comp1 = comp1
        self.comp2 = comp2
        self.comp3 = comp3
        self.comp4 = comp4
        self.comp5 = comp5

    def newNurse(self):
        print("New Nurse added \n", self.__dict__)

def staff_comp(key_words):
    folderpath = r"C:\Python\Comp"
    filepaths = [os.path.join(folderpath, name) for name in os.listdir(folderpath)]
    comp_req = []
    num_count = 0
    with open("./data/compResults", 'w', encoding="utf-8") as r:
        for word in key_words:
            for path in filepaths:
                with open(path, 'r', encoding="utf-8") as f:
                    comps = json.load(f)
                    for comp in comps:
                        #print(word, "-->",comp, "\n")
                        if word == comp:
                            #print("Found competency match ", comps[0])
                            num_count = num_count+1
                            comp_req.append(comps[0])
    print("\nNumber of competencies found = ", num_count)
    return(comp_req)

def staff_allocation(staff, patient):
    score = 0

```



```

for comp in patient:
    if comp == staff.comp1:
        print("Staff ", staff.ID, " has appropriate competency for patient ", staff.comp1)
        score = score+1
    if comp == staff.comp2:
        print("Staff ", staff.ID, " has appropriate competency for patient ", staff.comp2)
        score = score+1
    if comp == staff.comp3:
        print("Staff ", staff.ID, " has appropriate competency for patient ", staff.comp3)
        score = score+1
    if comp == staff.comp4:
        print("Staff ", staff.ID, " has appropriate competency for patient ", staff.comp4)
        score = score+1
    if comp == staff.comp5:
        print("Staff ", staff.ID, " has appropriate competency for patient ", staff.comp5)
        score = score+1
if score == len(patient):
    print("\nPatient allocated to staff", staff.ID)
    return(True)
else:
    print("Staff member does not have all required competencies, please review...")
    return(False)

```

G.2 *TF-IDF.py* Term Frequency – Inverse Document Frequency NLP Model

```

import os
from textwrap import indent
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans
from sklearn.metrics import adjusted_rand_score
import string
import numpy as np
from nltk.corpus import stopwords
from staff import Nurse
from staff import staff_comp
from staff import staff_allocation
import json

def load_data_Notes():
    folderpath = r"C:\Python\Notes"
    filepaths = [os.path.join(folderpath, name) for name in os.listdir(folderpath)]
    allfiles = ""

    for path in filepaths:
        with open(path, 'r', encoding="utf-8") as f:
            file = f.read()

```

```

        allfiles = allfiles + "\n\n" + file
    return (allfiles)

def load_file(file):
    with open(file, "r", encoding = "utf-8") as f:
        data = f.read()
    return(data)

def convert_json(file):
    word = file.replace("\n", " ")
    write_data("./data/NurseNotes.json", word)

def load_data(file):
    with open(file, "r", encoding = "utf-8") as f:
        data = json.load(f)
    return(data)

def save_data(file, data, staff, match, word_keys):
    with open(file, "w", encoding = "utf-8") as f:
        f.write("Key words found in progress note\n\n")
        word_keys = json.dump(word_keys, f, indent=4)
        f.write("\nPatient care requirements\n")
        data = json.dump(data, f, indent = 4)
        f.write("\n\n")
        f.write("Staff competencies\n")
        staff = json.dump(staff, f, indent = 4)
        f.write("\nStaff has appropriate competencies for Patient = ")
        match = json.dump(match, f, indent = 4)

def save_data_2(path, data, filename):
    with open(filename, "w", encoding="utf-8") as f:
        f.write("Hits from ")
        path = json.dump(path, f, indent=4)
        f.write("\n\n")
        data = json.dump(data, f, indent=4)

def write_data(file, data):
    with open (file, "w", encoding="utf-8") as f:
        json.dump(data, f, indent=4)

def remove_stops(text, stops):
    text = text.replace("\n", ' ')
    text = text.split(" ")
    #print(text)
    final = []
    for word in text:
        #print(word)
        if word not in stops:

```

```

        final.append(word)
    final = " ".join(final)
    final = final.translate(str.maketrans("", "", string.punctuation))
    while " " in final:
        final = final.replace(" ", " ")
    return(final)

def clean_docs(docs):
    stops = stopwords.words("english")
    final = []
    clean_doc = remove_stops(docs, stops)
    final.append(clean_doc)
    #print(final)
    return (final)

def progress_note(note1):
    cleaned_docs = clean_docs(note1)

    vectorizer = TfidfVectorizer(
        lowercase = True,
        max_features=40,
        max_df=1.0,
        min_df=0.0,
        ngram_range=(1,3),
        stop_words="english"
    )

    vectors = vectorizer.fit_transform(cleaned_docs)
    feature_names = np.array(vectorizer.get_feature_names())
    dense = vectors.todense()
    denselist = dense.tolist()
    all_keywords = []

    for data in denselist:
        x=0
        keywords = []
        for word in data:
            if word > 0:
                keywords.append(feature_names[x])
            x=x+1
        all_keywords.append(keywords)
    return(all_keywords)

staff_members = []
n1 = Nurse("1", "Registered Nurse", "Nursing Home", "Critical Care", "", "Wound Management",
           "Basic Life Support", "Chest Pain Management", "STEMI Management")
staff_members.append(n1)

```

```

n2 = Nurse("2", "Clinical Nurse", "Surgical Nursing", "Team Leading", "",
    "Advanced life Support", "Blood Sampling and Cannulation", "Arterial Sheath Removal", "Inotropic
Management" )
staff_members.append(n2)
n3 = Nurse("3", "Registered Nurse", "ICU Nurse", "Palliative", "TR Band", "Advanced Life Support", "12 Lead
ECG", "PICC Line", "Inotropic Management")
staff_members.append(n3)
n4 = Nurse("3", "Registered Nurse", "ICU Nurse", "Palliative", "Pacing Wire Removal", "Epidural and Regional
Block Pain", "Thrombolysis Management", "Basic Life Support", "Advanced Life Support")
staff_members.append(n4)
n1.newNurse()

```

```

folderpath = r"C:\Python\Notes"
filepaths = [os.path.join(folderpath, name) for name in os.listdir(folderpath)]
n=0
for path in filepaths:
    n=n+1
    print(path)
    with open(path, 'r', encoding="utf-8") as f:
        text = f.read()
        hits = progress_note(text)
        #print(hits)
        filename = "./TF-IDF/hits_data/TF_hits_N"+str(n)+".json"
        save_data_2(path, hits, filename)

```

```

patient = staff_comp(hits)
#print("Patient has the following care requirements ", patient)
match = staff_allocation(n1, patient)
staff_comp_list = []
staff_comp_list.append(n1.comp1)
staff_comp_list.append(n1.comp2)
staff_comp_list.append(n1.comp3)
staff_comp_list.append(n1.comp4)
staff_comp_list.append(n1.comp5)
save_data("./TF-IDF/allocation_data/tf-idf_S1N"+str(n)+".json", patient, staff_comp_list, match, hits)

```

```

match = staff_allocation(n2, patient)
staff_comp_list = []
staff_comp_list.append(n2.comp1)
staff_comp_list.append(n2.comp2)
staff_comp_list.append(n2.comp3)
staff_comp_list.append(n2.comp4)
staff_comp_list.append(n2.comp5)

```

```

save_data("./TF-IDF/allocation_data/tf-idf_S2N"+str(n)+".json", patient, staff_comp_list, match, hits)

match = staff_allocation(n3, patient)
staff_comp_list = []
staff_comp_list.append(n3.comp1)
staff_comp_list.append(n3.comp2)
staff_comp_list.append(n3.comp3)
staff_comp_list.append(n3.comp4)
staff_comp_list.append(n3.comp5)
save_data("./TF-IDF/allocation_data/tf-idf_S3N"+str(n)+".json", patient, staff_comp_list, match, hits)

match = staff_allocation(n4, patient)
staff_comp_list = []
staff_comp_list.append(n4.comp1)
staff_comp_list.append(n4.comp2)
staff_comp_list.append(n4.comp3)
staff_comp_list.append(n4.comp4)
staff_comp_list.append(n4.comp5)
save_data("./TF-IDF/allocation_data/tf-idf_S4N"+str(n)+".json", patient, staff_comp_list, match, hits)

```

G.3 *NMF.py* – Non-Negative Matrix NLP Model

```

import os
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans
from sklearn.metrics import adjusted_rand_score
from sklearn.decomposition import NMF
import string
import numpy as np
from nltk.corpus import stopwords
import json
from staff import *

def get_data(filename):
    with open(filename, "r", encoding = "utf-8") as f:
        data = f.read()
    data = data.lower()
    return (data)

def load_data(file):
    with open(file, "r", encoding = "utf-8") as f:
        data = json.load(f)
    return(data)

def write_data(file, data):
    with open (file, "w", encoding="utf-8") as f:

```

```

    json.dump(data, f, indent=4)

def convert_json(file):
    text = file.split("\n\n")
    write_data("./data/Notes.json", text)

def save_data(file, data, staff, match, word_keys):
    with open(file, "w", encoding = "utf-8") as f:
        f.write("Key words found in progress note\n\n")
        word_keys = json.dump(word_keys, f, indent=4)
        f.write("\nPatient care requirements\n")
        data = json.dump(data, f, indent = 4)
        f.write("\n\n")
        f.write("Staff competencies\n")
        staff = json.dump(staff, f, indent = 4)
        f.write("\nStaff has appropriate competencies for Patient = ")
        match = json.dump(match, f, indent = 4)

def save_data_2(path, data, filename):
    with open(filename, "w", encoding="utf-8") as f:
        f.write("Hits from ")
        path = json.dump(path, f, indent=4)
        f.write("\n\n")
        data = json.dump(data, f, indent=4)

def remove_stops(text, stops):
    #text = re.sub(r"[0-9][0-9]|\|\/| |[0-9]", "", text)
    text = text.replace("\n", ' ')
    text = text.split(" ")
    final = []
    for word in text:
        if word not in stops:
            final.append(word)
    final = " ".join(final)
    final = final.translate(str.maketrans("", "", string.punctuation))
    #final = "".join([i for i in final if not i.isdigit()])
    while " " in final:
        final = final.replace(" ", "")
    return(final)

def clean_docs(docs):
    stops = stopwords.words("english")
    final = []
    #for doc in docs:
    clean_doc = remove_stops(docs, stops)
    final.append(clean_doc)
    return (final)

```

```

def plot_top_words(model, feature_names, n_top_words, title):
    fig, axes = plt.subplots(2, 2, figsize=(10, 6), sharex=True)
    axes = axes.flatten()
    for topic_idx, topic in enumerate(model.components_):
        top_features_ind = topic.argsort()[::-n_top_words - 1:-1]
        top_features = [feature_names[i] for i in top_features_ind]
        weights = topic[top_features_ind]

        ax = axes[topic_idx]
        ax.barh(top_features, weights, height=0.7)
        ax.set_title(f'Topic {topic_idx + 1}',
                    fontdict={'fontsize': 8})
        ax.invert_yaxis()
        ax.tick_params(axis='both', which='major', labelsize=8)
        for i in 'top right left'.split():
            ax.spines[i].set_visible(False)
        fig.suptitle(title, fontsize=12)

plt.subplots_adjust(top=0.90, bottom=0.05, wspace=0.90, hspace=0.3)
plt.show()

def progress_note(note):
    cleaned_docs = clean_docs(note)
    n_top_words = 5

    vectorizer = TfidfVectorizer(
        lowercase = True,
        max_features=40,
        max_df=1.0,
        min_df=0,
        ngram_range=(1,3),
        stop_words="english"
    )
    vectors = vectorizer.fit_transform(cleaned_docs)
    feature_names = np.array(vectorizer.get_feature_names())
    dense = vectors.todense()
    denselist = dense.tolist()

    nmf = NMF(n_components=4, solver="mu", beta_loss='frobenius')
    W = nmf.fit_transform(vectors)
    H = nmf.components_

    final_topics = []
    for i, topic in enumerate(H):
        print([str(x) for x in feature_names[topic.argsort()[-5:]]])
        final_topics.append([str(x) for x in feature_names[topic.argsort()[-5:]]])
    final_comp = []
    for x in final_topics:

```

```

for key in x:
    final_comp.append(key)
return(final_comp)

```

```

staff_members = []
n1 = Nurse("1", "Registered Nurse", "Nursing Home", "Critical Care", "", "Wound Management",
           "Basic Life Support", "Chest Pain Management", "STEMI Management")
n1.newNurse()
staff_members.append(n1)
n2 = Nurse("2", "Clinical Nurse", "Surgical Nursing", "Team Leading", "",
           "Advanced life Support", "Blood Sampling and Cannulation", "Arterial Sheath Removal", "12 Lead ECG"
)
#n2.newNurse()
staff_members.append(n2)
n3 = Nurse("3", "Registered Nurse", "ICU Nurse", "Palliative", "TR Band Management", "Advanced Life
Support", "12 Lead ECG", "PICC Line", "Arterial Sheath Removal")
#n3.newNurse()
staff_members.append(n3)
n4 = Nurse("3", "Registered Nurse", "ICU Nurse", "Palliative", "Pacing Wire Removal", "Epidural and Regional
Block Pain", "Thrombolysis Management", "Basic Life Support", "Advanced Life Support")
staff_members.append(n4)

```

```

folderpath = r"C:\Python\Notes"
filepaths = [os.path.join(folderpath, name) for name in os.listdir(folderpath)]
n=0
for path in filepaths:
    n=n+1
    print("\n", path, "\n")
    with open(path, 'r', encoding="utf-8") as f:
        text = f.read()
        hits = progress_note(text)
        #print(hits)
        filename = "./NMF/hits_data/NMF_hits_N"+str(n)+".json"
        save_data_2(path, hits, filename)

```

```

patient = staff_comp(hits)
#print("Patient has the following care requirements ", patient)
match = staff_allocation(n1, patient)
staff_comp_list = []
staff_comp_list.append(n1.comp1)
staff_comp_list.append(n1.comp2)
staff_comp_list.append(n1.comp3)
staff_comp_list.append(n1.comp4)
staff_comp_list.append(n1.comp5)
save_data("./NMF/allocation_data/NMF_S1N"+str(n)+".json", patient, staff_comp_list, match, hits)

```



```

match = staff_allocation(n2, patient)
staff_comp_list = []
staff_comp_list.append(n2.comp1)
staff_comp_list.append(n2.comp2)
staff_comp_list.append(n2.comp3)
staff_comp_list.append(n2.comp4)
staff_comp_list.append(n2.comp5)
save_data("./NMF/allocation_data/NMF_S2N"+str(n)+".json", patient, staff_comp_list, match, hits)

```

```

match = staff_allocation(n3, patient)
staff_comp_list = []
staff_comp_list.append(n3.comp1)
staff_comp_list.append(n3.comp2)
staff_comp_list.append(n3.comp3)
staff_comp_list.append(n3.comp4)
staff_comp_list.append(n3.comp5)
save_data("./NMF/allocation_data/NMF_S3N"+str(n)+".json", patient, staff_comp_list, match, hits)

```

```

match = staff_allocation(n4, patient)
staff_comp_list = []
staff_comp_list.append(n4.comp1)
staff_comp_list.append(n4.comp2)
staff_comp_list.append(n4.comp3)
staff_comp_list.append(n4.comp4)
staff_comp_list.append(n4.comp5)
save_data("./NMF/allocation_data/NMF_S4N"+str(n)+".json", patient, staff_comp_list, match, hits)

```

G.5 *Multiclass.py* Multiclass text classification Model

```

from numpy import array
from keras.preprocessing.text import one_hot
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers.core import Activation, Dropout, Dense
from keras.layers import Flatten, LSTM
from keras.layers import GlobalMaxPooling1D
from keras.models import Model
from keras.layers.embeddings import Embedding
from numpy.core.fromnumeric import argmax
from numpy.lib.arraypad import pad
from sklearn.model_selection import train_test_split
from keras.preprocessing.text import Tokenizer
from keras.layers import Input
from keras.layers.merge import Concatenate
import tensorflow as tf
from tensorflow.keras.layers import TextVectorization

```

```

import io
import shutil
import string

import pandas as pd
from sklearn.utils import shuffle
import numpy as np
from numpy import array
from numpy import asarray
from numpy import zeros
import re
import os
import json

import matplotlib.pyplot as plt

def get_data(filename):
    with open(filename, "r", encoding = "utf-8") as f:
        data = f.read()
        data = data.lower()
    return (data)

def convert_json(file):
    text = file.split("\n\n")
    write_data("./Multi_data/Pt_notes_1.json", text)

def write_data(file, data):
    with open (file, "w", encoding="utf-8") as f:
        json.dump(data, f, indent=4)

def load_data_Notes():
    folderpath = r"C:\Python\Multi_Comp"
    filepaths = [os.path.join(folderpath, name) for name in os.listdir(folderpath)]
    allfiles = []

    for path in filepaths:
        with open(path, 'r', encoding="utf-8") as f:
            file = f.read()
            file = preprocess_text(file)
            file = file.lower()
            allfiles.append(file)
    return (allfiles)

def load_data(file):
    with open(file, "r", encoding = "utf-8") as f:

```

```

    data = json.load(f)
    return(data)

def preprocess_text(sen):
    # Remove punctuations and numbers
    sentence = re.sub('[^a-zA-Z]', '', sen)

    # Single character removal
    sentence = re.sub(r"\s+[a-zA-Z]\s+", '', sentence)

    # Removing multiple spaces
    sentence = re.sub(r'\s+', ' ', sentence)

    return sentence

def create_index(texts, filename):
    words = texts.split()
    tokenizer = Tokenizer(num_words=100000)
    tokenizer.fit_on_texts(words)
    sequences = tokenizer.texts_to_sequences(words)
    word_index = tokenizer.word_index
    print(f"Found {len(word_index)} unique words.")
    with open (filename, "w") as f:
        json.dump(word_index,f, indent=4)

def label_data(sentences, label):
    total_chunks = []
    for sentence in sentences:
        total_chunks.append((sentence, label))
    return(total_chunks)

competency_labels = pd.read_csv('./Multi_data/Comp_label.csv')
print(competency_labels)

competency_labels.sum(axis=1).plot.bar()

competencies = competency_labels[["12 Lead ECG", "Wound Management", "Arterial Sheath Removal",
"Basic Life Support", "Inotropic Management", "TR Band Management"]]
competencies.head()
X = []
X = load_data_Notes()
y = competencies.values[1:601]

y = shuffle(y)

```

```

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state=42)

tokenizer = Tokenizer(num_words=5000)
tokenizer.fit_on_texts(X_train)

X_train = tokenizer.texts_to_sequences(X_train)
print(X_train)
X_test = tokenizer.texts_to_sequences(X_test)

vocab_size = len(tokenizer.word_index)+1

maxlen=250

X_train = pad_sequences(X_train, padding='post', maxlen=maxlen)
X_test = pad_sequences(X_test, padding='post', maxlen=maxlen)

embeddings_dictionary = dict()

glove_file = open('./Multi_data/glove.6B/glove.6B.300d.txt', encoding="utf8")

for line in glove_file:
    records = line.split()
    word = records[0]
    vector_dimensions = asarray(records[1:], dtype='float32')
    embeddings_dictionary[word] = vector_dimensions
glove_file.close()

embedding_matrix = zeros((vocab_size, 300))
for word, index in tokenizer.word_index.items():
    embedding_vector = embeddings_dictionary.get(word)
    if embedding_vector is not None:
        embedding_matrix[index] = embedding_vector
    print(embedding_matrix)

deep_inputs = Input(shape=(maxlen,))
embedding_layer = Embedding(vocab_size, 300, weights=[embedding_matrix],
trainable=False)(deep_inputs)
LSTM_Layer_1 = LSTM(512)(embedding_layer)
dense_layer_1 = Dense(6, activation='softmax')(LSTM_Layer_1)
model = Model(inputs=deep_inputs, outputs=dense_layer_1)

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['acc'])

print(model.summary())

history = model.fit(X_train, y_train, batch_size=16, epochs=8, verbose=1, validation_split=0.3)

score = model.evaluate(X_test, y_test, verbose=1)

```

```

model.save("./Multi_data/AllocationModel")

print("Test Score:", score[0])
print("Test Accuracy:", score[1])

a = model.predict(X_test)
print(np.argmax(a))

import matplotlib.pyplot as plt

#plt.plot(history.history['acc'])
#plt.plot(history.history['val_acc'])

plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])

plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

```

G.6 AllocationML.py Multiclass text classification model validation

```

import re
import os
import json
import tensorflow as tf
from tensorflow import keras
from keras.preprocessing.sequence import pad_sequences
from keras.preprocessing.text import Tokenizer

def get_data(filename):
    with open(filename, "r", encoding = "utf-8") as f:
        data = f.read()
    data = data.lower()
    return (data)

```

```

def convert_json(file):
    text = file.split("\n\n")
    write_data("./Multi_data/Pt_notes_1.json", text)

def write_data(file, data):
    with open (file, "w", encoding="utf-8") as f:
        json.dump(data, f, indent=4)

def load_data_Notes():
    folderpath = r"C:\Python\Multi_Comp"
    filepaths = [os.path.join(folderpath, name) for name in os.listdir(folderpath)]
    allfiles = []

    for path in filepaths:
        with open(path, 'r', encoding="utf-8") as f:
            file = f.read()
            file = preprocess_text(file)
            file = file.lower()
            allfiles.append(file)
    return (allfiles)

def load_data(file):
    with open(file, "r", encoding = "utf-8") as f:
        data = json.load(f)
    return(data)

def preprocess_text(sen):
    # Remove punctuations and numbers
    sentence = re.sub('[^a-zA-Z]', '', sen)

    # Single character removal
    sentence = re.sub(r"\s+[a-zA-Z]\s+", '', sentence)

    # Removing multiple spaces
    sentence = re.sub(r'\s+', ' ', sentence)

    return sentence

data = get_data("./Notes/NurseNote7.txt")
data = preprocess_text(data)
print(data)
#data = load_data_Notes()

tokenizer = Tokenizer(num_words=5000)
tokenizer.fit_on_texts(data)

```

```

data = tokenizer.texts_to_sequences(data)
Note_res = pad_sequences(data, padding='post', maxlen=250)

new_model = tf.keras.models.load_model('./Multi_data/AllocationModel')
new_model.summary()
test = new_model.predict(Note_res)
print(test)

```

G.7 *SpacyModelTest.py* Spacy NER NLP Model

```

from textwrap import indent
import spacy
import json
import os
from spacy.lang.en import English
from spacy.pipeline import EntityRuler
from spacy.lang.en import STOP_WORDS
from staff import Nurse
from staff import staff_comp
from staff import staff_allocation

def load_data(file):
    with open(file, "r", encoding = "utf-8") as f:
        data = json.load(f)
    return(data)

def save_data(file, data, staff, match, word_keys):
    with open(file, "w", encoding = "utf-8") as f:
        f.write("Key words found in progress note\n\n")
        word_keys = json.dump(word_keys, f, indent=4)
        f.write("\nPatient care requirements\n")
        data = json.dump(data, f, indent = 4)
        f.write("\n\n")
        f.write("Staff competencies\n")
        staff = json.dump(staff, f, indent = 4)
        f.write("\nStaff has appropriate competencies for Patient = ")
        match = json.dump(match, f, indent = 4)

def save_data_2(path, data, filename):
    with open(filename, "w", encoding="utf-8") as f:
        f.write("Hits from ")

```

```

path = json.dump(path, f, indent=4)
f.write("\n\n")
data = json.dump(data, f, indent=4)

```

```

def Nurses(file):
    data = load_data(file)
    staff_list = []
    for item in data:
        names = item.split(" ")
        for name in names:
            name = name.strip()
            staff_list.append(name)
        staff_list.sort()

```

```

return staff_list

```

```

def create_training_data(file, type):
    data = Nurses(file)
    patterns = []
    for item in data:
        pattern = {
            "label": type,
            "pattern": item
        }
        patterns.append(pattern)
    return(patterns)

```

```

def gen_rules(pattern):
    nlp = English()
    ruler = nlp.add_pipe("entity_ruler")
    ruler.add_patterns(pattern)
    nlp.to_disk("nurse_NER")

```

```

def clean_data(data):
    doc = nlp(data)
    cleaned = []
    for token in doc:
        if not (token.is_stop):
            cleaned.append(token)
    return " ".join([str(item) for var in cleaned for item in var])
#print(cleaned)

```

```

def test_model(model, text):
    doc = model(text)
    results = []
    for ent in doc.ents:
        results.append(ent.text)

```



```

return results

def progress_note(note):

    segments = note.split("\n\n")
    hits = []
    for segment in segments:
        #print(segment)
        segment = segment.strip()
        segment = segment.replace("\n", " ")
        #print(segment)
        punc = "'!()-[]{};:'\"<>./?@#\$%^&* _~'"
        for ele in segment:
            if ele in punc:
                segment = segment.replace(ele, "")
                results = test_model(nlp, segment)

        for result in results:
            print(result)
            if result not in STOP_WORDS:
                hits.append(result)

    print(hits)
    return(hits)

patterns = create_training_data("./key_doc.json", "COMP")
gen_rules(patterns)

nlp = spacy.load("nurse_NER") ##Nurse_ner_model

staff_members = []
n1 = Nurse("1", "Registered Nurse", "Nursing Home", "Critical Care", "", "Wound Management",
           "Basic Life Support", "Chest Pain Management", "STEMI Management")
n1.newNurse()
staff_members.append(n1)
n2 = Nurse("2", "Clinical Nurse", "Surgical Nursing", "Team Leading", "",
           "Advanced life Support", "Blood Sampling and Cannulation", "Arterial Sheath Removal", "12 Lead ECG"
)
#n2.newNurse()
staff_members.append(n2)
n3 = Nurse("3", "Registered Nurse", "ICU Nurse", "Palliative", "TR Band Management", "Advanced Life
Support", "12 Lead ECG", "PICC Line", "Arterial Sheath Removal")
#n3.newNurse()
staff_members.append(n3)

n4 = Nurse("4", "Registered Nurse", "ICU Nurse", "Palliative", "Pacing Wire Removal", "Epidural and Regional
Block Pain", "Thrombolysis Management", "Basic Life Support", "Advanced Life Support")

```

```
staff_members.append(n4)
```

```
folderpath = r"C:\Python\Notes"
```

```
filepaths = [os.path.join(folderpath, name) for name in os.listdir(folderpath)]
```

```
n=0
```

```
for path in filepaths:
```

```
    n=n+1
```

```
    print(path)
```

```
    with open(path, 'r', encoding="utf-8") as f:
```

```
        text = f.read()
```

```
        hits = progress_note(text)
```

```
        filename = "./Spacy/hits_data/SP_hits_N"+str(n)+".json"
```

```
        save_data_2(path, hits, filename)
```

```
    patient = staff_comp(hits)
```

```
    #print("Patient has the following care requirements ", patient)
```

```
    match = staff_allocation(n1, patient)
```

```
    staff_comp_list = []
```

```
    staff_comp_list.append(n1.comp1)
```

```
    staff_comp_list.append(n1.comp2)
```

```
    staff_comp_list.append(n1.comp3)
```

```
    staff_comp_list.append(n1.comp4)
```

```
    staff_comp_list.append(n1.comp5)
```

```
    save_data("./Spacy/allocation_data/SpaCy_S1N"+str(n)+".json", patient, staff_comp_list, match, hits)
```

```
    match = staff_allocation(n2, patient)
```

```
    staff_comp_list = []
```

```
    staff_comp_list.append(n2.comp1)
```

```
    staff_comp_list.append(n2.comp2)
```

```
    staff_comp_list.append(n2.comp3)
```

```
    staff_comp_list.append(n2.comp4)
```

```
    staff_comp_list.append(n2.comp5)
```

```
    save_data("./Spacy/allocation_data/SpaCy_S2N"+str(n)+".json", patient, staff_comp_list, match, hits)
```

```
    match = staff_allocation(n3, patient)
```

```
    staff_comp_list = []
```

```
    staff_comp_list.append(n3.comp1)
```

```
    staff_comp_list.append(n3.comp2)
```

```
    staff_comp_list.append(n3.comp3)
```

```
    staff_comp_list.append(n3.comp4)
```

```
    staff_comp_list.append(n3.comp5)
```

```
    save_data("./Spacy/allocation_data/SpaCy_S3N"+str(n)+".json", patient, staff_comp_list, match, hits)
```

```
    match = staff_allocation(n4, patient)
```

```
    staff_comp_list = []
```

```
    staff_comp_list.append(n4.comp1)
```

```

staff_comp_list.append(n4.comp2)
staff_comp_list.append(n4.comp3)
staff_comp_list.append(n4.comp4)
staff_comp_list.append(n4.comp5)
save_data("./Spacy/allocation_data/SpaCy_S4N"+str(n)+".json", patient, staff_comp_list, match, hits)

```

G.6 *data_gen.py* Alternate data creation script

```

import os
import spacy
import json
import random
from nltk.corpus import stopwords
from scipy.stats import uniform
import string

def load_data(file):
    with open(file, "r", encoding = "utf-8") as f:
        data = f.read()
    return(data)

def convert_json(file):
    text = file.split("\n\n")
    text = file.split(" ")
    write_json("./data/Note.json", text)

def write_json(file, data):
    with open(file, "w", encoding="utf-8") as f:
        json.dump(data, f, indent=4)

def write_data(file, data):
    with open (file, "w", encoding="utf-8") as f:
        f.write(data)

def load_json(file):
    with open(file, "r", encoding = "utf-8") as f:
        data = json.load(f)
    return(data)

```

```

def random_data(file, terms, key_doc):
    num = 0
    t = 0
    new_note = ""
    for i in range(len(file)):
        dist = []
        ran = random.randint(1, 10)
        dist.append(ran)
        #print(dist)
        num = random.randint(1,98120)
        re_term = terms[num]
        t = t+ran
        if t < len(file):
            word = file[t]
            if word not in key_doc:
                file[t] = word.replace(word, re_term)
        else:
            break
    for word in file:
        new_note = new_note + " " + word
    return(new_note)

def random_num():
    n = 10
    start = 1
    width = 11
    data_uniform = uniform.rvs(size=n, loc = start, scale=width)
    print(data_uniform)
    return(data_uniform)

data = load_data("./Notes/NurseNote3.txt")
medterm = load_json("./data/medTerm.json")
key_doc = load_json("./Comp/12lead.json")
convert_json(data)
ran_num = random_num()
#data = load_json("./data/Note.json")
for s in range(5):
    data = load_json("./data/Note.json")
    print("Generating random Pt_note data")
    new_note = random_data(data, medterm, key_doc)
    write_data("./Comp3/12lead_Pt%s.txt" % s, new_note)

```