Natural Language Processing as a Tool in Supporting Clinical Decision-Making

By

Laurence Robert Jones

A thesis submitted in partial fulfilment for the

degree of Doctor of Philosophy

in the

Faculty of Computing, Engineering and Science

University of South Wales

in collaboration with

Digital Health and Care Wales

NHS Wales

February 2023

Graduate School

Ysgol Graddedigion

Candidate declaration

This is to certify that, except where specific reference is made, the work described in this thesis is the result of my own research. Neither this thesis, nor any part of it, has been presented, or is currently submitted, in candidature for any other award at this or any other University.

Signed

Candidate

Date 02/02/2023.

Acknowledgements

The past three and a half years have been an endless cycle of stress, relief and personal growth that somehow resulted in a finished PhD thesis. It began with other people having the belief in me that I could accomplish what is contained in this thesis today, and I definitely wouldn’t have been able to complete it without their support.

Firstly, I have to thank Dr. Ian Wilson who has undoubtedly been the best supervisor anyone could ask for. Whether it was his professional input into the direction for the project, essentially teaching me to academically write from scratch, or endless Thursday afternoon chats about nothing related to the project itself, I could always rely on an honest (and mostly positive) view on how I was progressing. I would also like to thank my other supervisors Prof. Andrew Ware and Dr. Penny Holborn for the chats that we shared across the project, often introducing levity to otherwise bothersome situations.

A special thanks to all those at Digital Health and Care Wales (DHCW) who have assisted in the project, especially Gareth John, Dave Price, and Dr. Tim O’Sullivan who were integral in securing the data needed to make this project a success.

Last but not least I would like to thank all my family and friends who have been with me across this period and supported me each and every step of the way.

Abstract

While the amount of unstructured text data continues to grow within the clinical domain, little modelling is carried out in comparison to other industries. My research goal in this thesis is to present machine learning models that can effectively discern the relationships within medical notes, tying symptoms and other elements to an associated medical speciality.

There have been many studies in the clinical domain using natural language processing that have seen successes with document classification. However, the solutions proposed often rely on an external medical dictionary to annotate the data. My goal is the development of a classifier that shows that these relationships can be extracted from the original, unstructured text. Furthermore, the standard approach to documenting research in this area revolves around focusing on a single type of machine learning algorithm, be it the method of feature generation or the specific machine learning model chosen for the task. The results shown in this thesis address this issue by providing a comparative demonstration of multiple feature generation methods alongside a plethora of traditional machine learning and neural network-based models for classification. Lastly, existing research encounters issues with the procurement of suitable medical data, often defaulting to using datasets that have been curated for a specific task. This research instead uses real patient data from Digital Health and Care Wales (DHCW), selected randomly from cases between 2018 and 2019.

The results produced in this thesis found that frequency-based feature generation performed substantially better than word embeddings when using a traditional machine learning model like logistic regression. However, using word embeddings with a neural network architecture yielded more comparable results. For the machine learning models themselves, the support vector machine (91%) and two transformer deep learning models (93%) produced the best results.

Table of Contents

Table of Figures

Table of Tables

Copyright Declaration

The author hereby gives consent for this thesis to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Chapter 1 Introduction

Natural language processing is a subfield of computer science that specialises in marrying linguistics with statistical methodologies and artificial intelligence. Whilst research into statistical modelling of language data has been around since the 1980s, the past twenty years have seen an upsurge in interest in exploiting such data across a great many sectors, including business, law, and healthcare. This interest originates from the increased ability to perform data driven decision making with the modern availability of both data and the required hardware. However, as algorithms and techniques used within Natural Language Processing are rapidly evolving, there is conflicting research into which techniques are considered best practices. This, coupled with the fact that a dataset is usually built for a purpose and is therefore significantly different from another dataset, presents a unique issue for generalisation. This thesis aims to evaluate available approaches for performing document classification with the intent of helping medical professionals by supplying insight through a knowledge repository.

This chapter provides an introductory overview of the study's context and wherein the research problem lies, followed by the research aims, objectives and questions alongside the significance and limitations of the study.

1.1 Problem Statement

The provision of accurate information between clinicians is a necessity for safeguarding patient health. Existing systems like the NHS Wales Clinical Portal provide consultants with electronic access to a patient's entire medical record when making treatment decisions during a referral process as opposed to a single handwritten note by the patient's general practitioner. It is accepted that medicine is a complex discipline, requiring years of training before a member of the public can become a practicing clinician. Yet, even with such training, clinicians may need to incorporate additional materials like medical dictionaries, past experiences, and a consultation with another clinician to ensure the correct decision is made to best meet a patient's needs. When looking specifically at patient transferrals, this complexity can cause differences in opinions between the two clinicians involved. Each party will have a separate set of information influenced by the general practitioner being the only one having direct communication with the patient.

Studies have shown that machine learning techniques can be incorporated into the clinical domain with reasonable success. Research has been carried out in natural language using patient narratives (Buchan, et al., 2017; Munkhdalai, et al., 2018) for document classification and in radiology reports for image recognition (Sevenster, et al., 2015; Zech, et al., 2018). However, there are underlying issues with existing research in this field. Firstly, due to the nature of the data in use, it is not always possible to access a suitable dataset. An example of this would be Hassanpour, Bay, & Langlotz (2017) in which the dataset of radiology reports only numbered forty documents. This thesis combats this by working on three intrinsically different datasets of sizes. A presenting complaints dataset of 839,330, a urology specific dataset of 25,451, and a general referral dataset numbering 111,128.

Another issue surrounding existing research into machine learning within the clinical domain is the specificity of the research targets. Even when looking at benchmarking challenges, the goal is to create a system that can decide whether the clinical data refers to one of very few classes (e.g., smoker, past smoker or non-smoker) (McCormick, et al., 2008; Uzuner, et al., 2008; Wicentowski & Sydes, 2008). Alternatively, publications show that the created systems work for a rigid, structured dataset that includes a document and additional markers for patient biometrics that can help direct the system to its goal.

Many of the research publications feature the inclusion of a pre-existing named entity system like cTAKES (clinical Text Analysis and Knowledge Extraction System, (Savova, et al., 2010)) or MetaMap (Aronson, 2001) to convert examples of medical terms contained within documents to codes derived from the Unified Medical Language System (Bodenreider, 2004), an extensive medical data dictionary that contains other medical vocabularies like SNOMED Clinical Terms (Snomed International, 2022). Whilst there are newer attempts at document classification in the clinical domain without the inclusion of clinical ontologies (Hughes, et al., 2017), the issue with procuring data reappears wherein the research has to be completed by using structured, well-written clinical research papers gathered from PubMed rather than actual examples of full-bodied letters written by a clinical professional or a member of staff as used in this thesis. The affirmation of the potential to create a system without additional ontologies comes from recent examples in the clinical domain (Cohen, et al., 2016; Hughes, et al., 2017; Weng, et al., 2017). The research showed that factors including the presentation of the data to the system through word vector representations and the classifier chosen were far more influential on the accuracy of the system than the inclusion of any medical ontology (0.005% difference in accuracy between the system including the best curated ontological subset and the baseline model).

1.2 Research Aim and Objectives

The aim of this research is to show the effectiveness of document classification using medical notes without the inclusion of additional domain knowledge.

The intention was to create a system that could successfully discover which semantic elements (symptoms etc.) present within a doctors' note attributed to the assigned medical specialities and patient priorities. The resulting classifications could then exist as an additional information source to support clinicians.

In order to achieve this aim, the following objectives have been integrated into this work:

- To investigate machine learning and its existing use within the clinical domain alongside the fundamental decision making of clinicians when referring patients to hospital.
- To assess the impact that different feature generation techniques can have on the ability for a machine learning algorithm to model relationships in the data.
- To develop a natural language processing classification pipeline that takes in raw text data and outputs labels according to medical speciality or priority.
- To evaluate the contributing factors to a successful classification including feature generation, model selection and feature reduction.

1.3 Thesis outline

This PhD thesis consists of seven chapters and is structured as follows:

Chapter 2 provides an overview into the clinical referral process before introducing the concepts of Natural Language Processing and the associated classification pipeline. It fulfils the first research objective by discussing the existing literature relevant to the project and the issues that can occur with medical datasets such as a lack of publicly available data.

Chapter 3 outlines different vectorisation methods that can be utilised to transform a dataset consisting of unedited medical documents into a machine-readable format. Furthermore, this chapter also shows how feature reduction techniques can take place during pre-processing to reduce the size of an overall dataset without removing significant variance.

Chapter 4 covers the setup for experiments including the hardware used to create, train, and implement the models presented in this research. This is followed by an explanation for the use of all the different libraries used in this research. The chapter then outlines the steps taken to evaluate and clean each of the datasets before modelling. The chapter then defines the different methods used for clustering and classification in this thesis including any associated changes in hyperparameters.

Chapter 5 presents all of the findings of this research across the three datasets. The initial findings using the presenting complaints dataset are discussed, which showed the potential for applying machine learning techniques to previously untapped data. Then, clustering and classification results are shown for the urology dataset. This section outlines the best vectorisation, classification and feature reduction methods discovered when interacting with this dataset including a threshold-based feature elimination method. The same approach is then taken with the general referral's dataset, wherein the accuracy of the models proved significantly higher on a general dataset than a specialised dataset including the latest transformer model specialised to this type of data. The final part of Chapter 5, 5.4, outlines the results that show that the goal of this research is valid. The support system indicates the top three classes associated with the classification with an accuracy of 99% and can offer the probabilities for each class to the end user.

Chapter 6 covers the main conclusions for this research and includes a discussion relating to each of the three research questions outlined in Chapter 1. The contributions to knowledge are then outlined based around the unique perspective that has been permitted with these datasets. Finally, the future work system outlines how the successes of the research could be translated into a live environment and the added steps that would need to be taken to ensure patient safety.

The appendices contain additional information relating to the datasets used in this research as well as some more detailed clustering and classification reports.

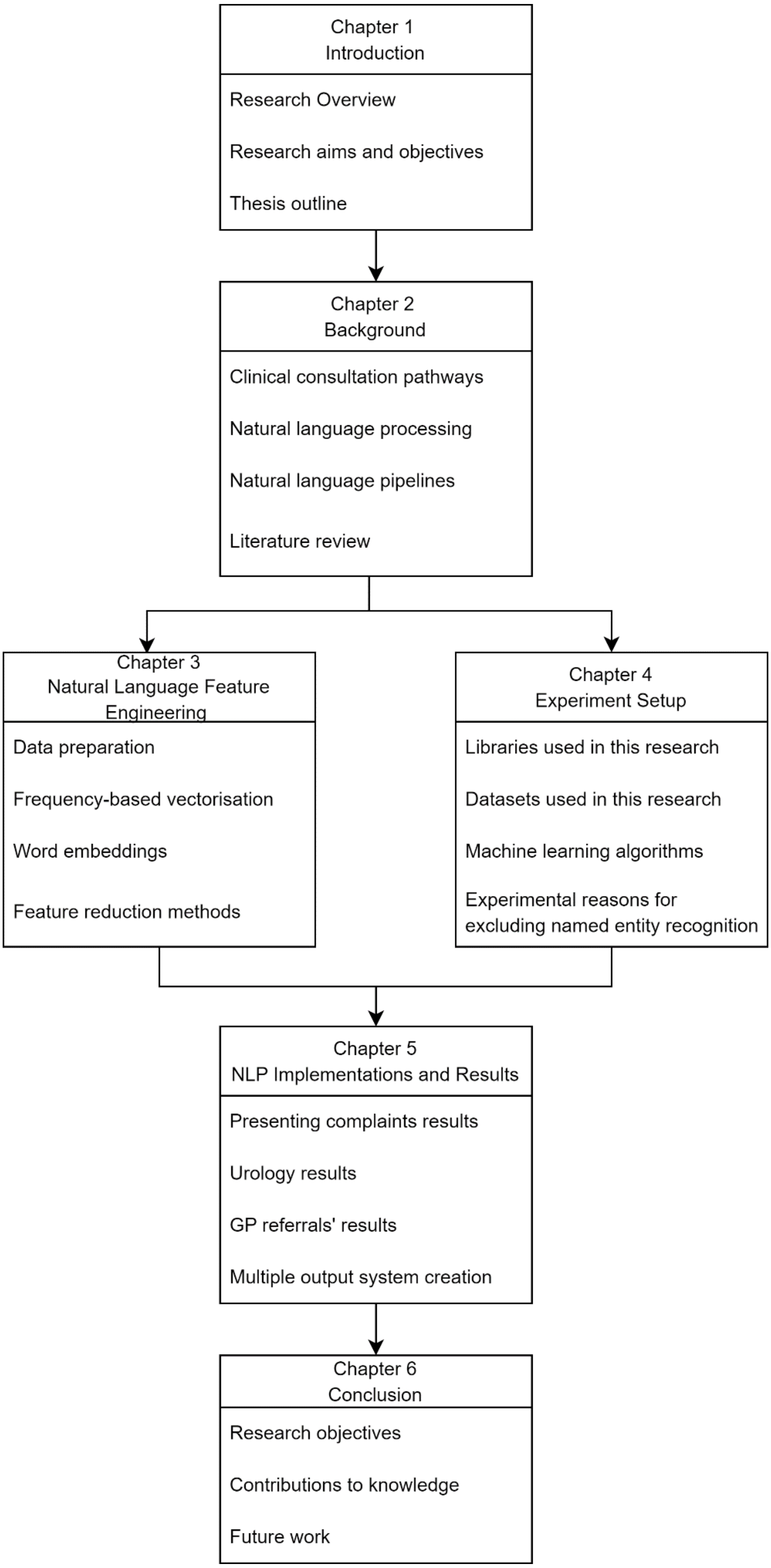

Chapter 1
Introduction
Research Overview
Research aims and objectives
Thesis outline
Chapter 2
Background
Clinical consultation pathways
Natural language processing
Natural language pipelines
Literature review
Chapter 3
Natural Language Feature Engineering
Data preparation
Frequency-based vectorisation
Word embeddings
Feature reduction methods
Chapter 4
Experiment Setup
Libraries used in this research
Datasets used in this research
Machine learning algorithms
Experimental reasons for excluding named entity recognition
Chapter 5
NLP Implementations and Results
Presenting complaints results
Urology results
GP referrals' results
Multiple output system creation
Chapter 6
Conclusion
Research objectives
Contributions to knowledge
Future work

Figure 1.1 Thesis Outline

Chapter 2 Background

This chapter provides background into the problem space of the research. It begins by outlining the existing manual process of referring a patient from general practitioner to consultant with insights into the issues surrounding clarification or missing information that may occur. After the clinical consultation pathway is discussed, an overview of Natural Language Processing is introduced including the pipeline method used to describe the steps taken to transform a set of unorganised data into usable documents and an eventual target outcome.

2.1 Clinical consultation pathways

Before considering the introduction of a system into an existing manual process it is important to consider the context of the situation. Within medicine, there are several pathways a patient may take over the course of their treatment. For instance, the initial treatment by a doctor or nurse in an accident & emergency department with a pathway to either a hospital admission or discharge. An example of the patient pathway discussed in this thesis is depicted in Figure 2.1. A patient will attend a consultation with a general practitioner for a medical issue and either be prescribed a course of treatment, sent for tests, or referred to a specialist. This referral may be to confirm a diagnosis, introduce specific treatment plans as well as giving specialist advice (Foot, et al., 2010).

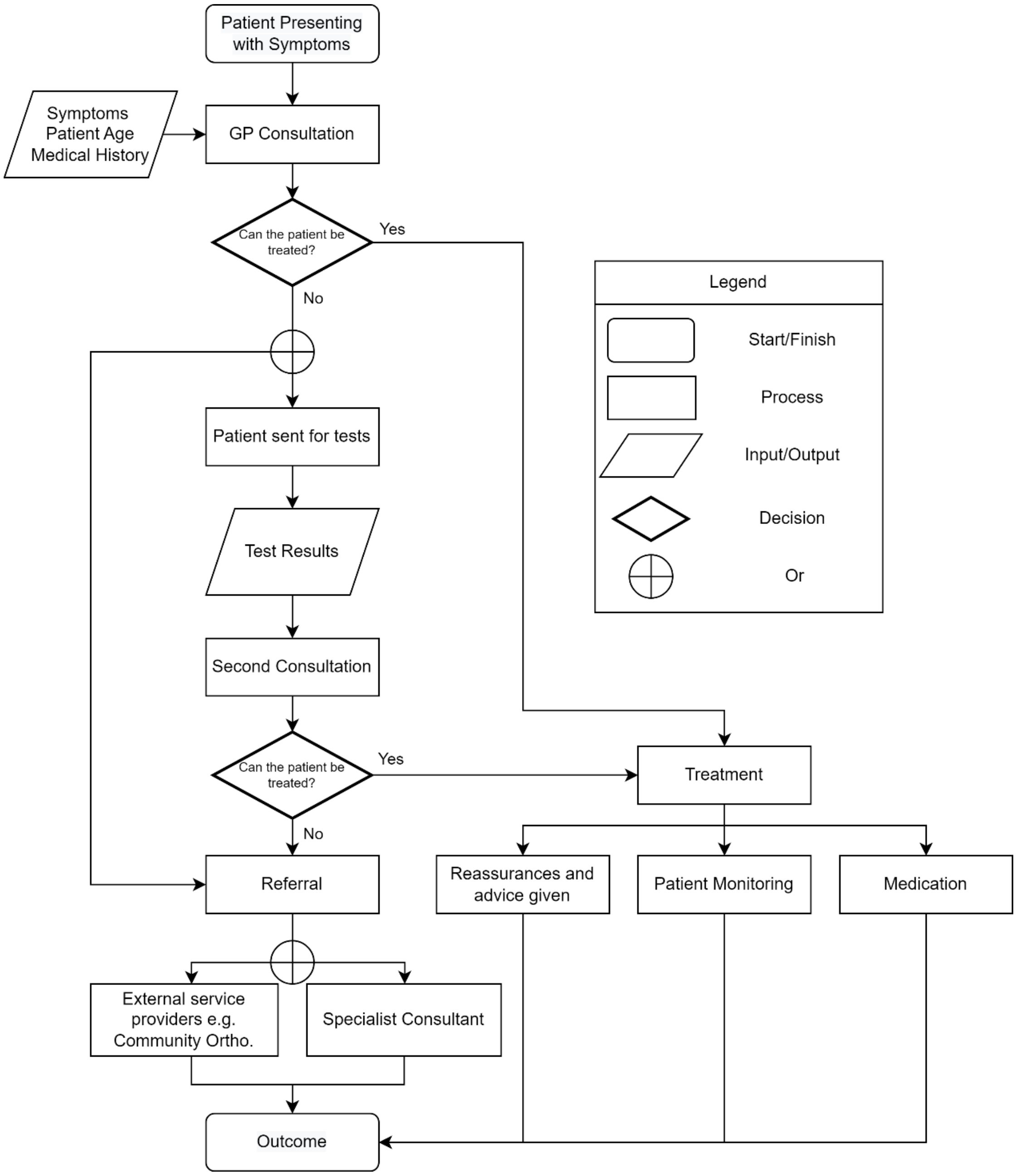


Figure 2.1 Patient pathway to specialist referral

The natural language aspect of the pathway this thesis is interested in comes during the specialist referral stage. A general practitioner in Wales will write a letter to a specialist via an electronic clinical portal (Welsh NHS Confederation, 2020) alongside patient biometrics and medical history. While the clinical portal has separate fields for patient attributes and medical history, experience dealing with the data used in this thesis and conversations with clinical professionals found that the important information is often duplicated within the main body of the letter alongside the information about the symptoms that the general practitioner has found or have been described by the patient.

Speaking with a specialist consultant, the level of detail contained within these letters is important for ensuring proper patient care as the consultant has no direct contact with the patient prior to the referral (NHS Wales, 2017). The referral letter needs to address any questions that the consultant may have regarding the case. Specifically:

- **Are the symptoms correct?**
    - There may be a situation where the patient describes ailments that cannot be triaged at a general practitioner's office. These ailments may be misinterpreted due to the language used by the patient. Occurrences of this issue may be mitigated by an associated medical history if available.
- **Is the priority assigned correctly?**
    - A consultant has no direct contact with the patient until the first consultation and relies on the language used by the general practitioner to convey the urgency of the patient's condition.
- **Is the patient being referred to the right specialist?**
    - Medicine is a complex science in which a series of symptoms may refer to one or more conditions. For example, this may occur in when the patient's age would affect the correct outcome; an elderly patient suffering from joint pain may need to be referred to either a rheumatologist (specialist in bones, muscles, and joints) or a specialist in geriatric medicine.

Should the specialist consultant have any concerns about the above three questions, the referral may be returned to the general practitioner for more clarification. This may be as simple as confirming a change in priority by the consultant or the request for more tests to be carried out (such as an electrocardiogram) before the consultant will agree that there is a need to see the patient in a specialist clinic. The time spent adjusting the referral conditions come at a loss of time till treatment for the patient and an increase in workload for both the general practitioner and the consultant.

2.2 Natural Language Processing

Natural language denotes a method of communication that has evolved organically alongside humanity, both written and verbal. Natural language systems build upon existing natural language sources to learn and infer from. These systems have become an integrated part of daily life, augmenting existing systems or replacing manual approaches, from digital assistants like Amazon’s Alexa to spam filtering and mobile text auto-predictions.

Natural language processing as a discipline describes the research and systems carried out to convert natural language data such as text or speech into a structured data format that can be manipulated by a computer system whilst creating an understandable output upon completion. These implementations are categorised into three major areas (Liu, et al., 2017). These areas are:

- Natural language generation refers to producing a human readable representation of a non-linguistic input (Reiter & Dale, 1997) such as text or speech (Gatt & Krahmer, 2018).
- Natural language understanding focuses on discovering the underlying links such as semantic or contextual markers, between a natural language input and a desired output (Semaan, 2012).
- Natural language interaction describes systems designed for direct human communication with a computer interface and receive a specific response, a process used in modern technologies such as online chatbots (Dale, 2016) .

Due to the nature of most Natural Language Processing tasks, it is commonplace to find that methodologies span across more than one branch of the discipline.

The theory that underpins this thesis and a great many other Natural Language Processing projects is the distributional hypothesis. The distributional hypothesis states that “a word is characterized by the company in which it keeps” (Firth, 1957), meaning terms that are similar to each other will appear in similar contexts. This hypothesis forms the basis for statistical semantics and later natural language processing techniques such as latent semantic analysis, clustering, and document classification. It is the idea that a document, an image, or a section of speech is a collection of terms that make up a whole and that whole can be compared with other collections of terms to determine semantic similarity.

However, written language is not constructed in such a way that each word is individually important and conveys the same meaning without its surrounding context. The example in Figure 2.2 shows that although most of the words are present in both sentences, which would satisfy a similarity measure like the Jaccard index, the lexical semantics of the sentences are completely different. The homonyms *green* and *leaves* transform the sentences depending on morphological structure and contextual markers. The approaches taken to alleviate this issue using word vector representations will be introduced and discussed in the Chapter 3 of the thesis.

The tree has green leaves.

The ball leaves the green.

Figure 2.2 Similar sentences that are semantically different

2.3 Natural language pipelines

A natural language processing pipeline outlines the different steps that need to be taken to transform a raw text input into a series of associated outputs. Whilst textual information contains rich knowledge that can be used for machine learning, the issue with working with natural language is that it is inherently noisy. As such, approaching a Natural Language Processing task is best viewed as a pipeline formed of several smaller sub-tasks each contributing to an overall goal.

Some of the issues facilitating the necessity of sub-tasks are items like common words present within natural language. Grammatical articles like ‘the’ and ‘an’ provide no potential learning opportunities to a machine learning model. Additionally, as humans we can understand that the length of a sentence is irrelevant and separate to the semantics contained within. However, when it comes to the input to a machine learning model for classification, such as a support vector machine or an artificial neural network, a computer expects all information to be presented with a fixed length (Mikolov, et al., 2013).

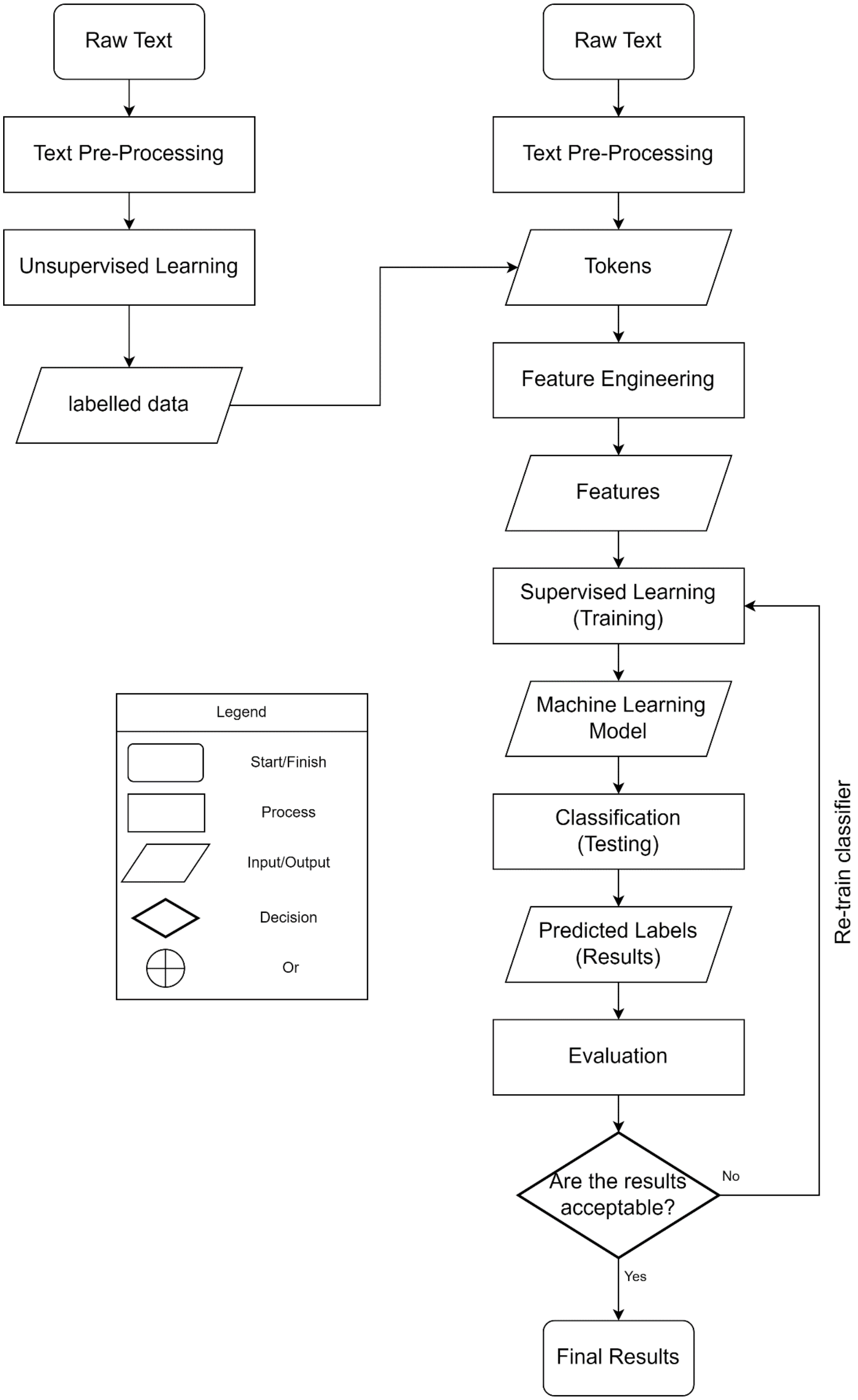


Figure 2.3 Classification pipeline including label generation.

A generalised outline of a natural language classification pipeline for text classification is shown in Figure 2.3. Each stage in the pipeline shows a step taken to modify the data itself and movement towards creating the final output.

The first data manipulation happens in the pre-processing stage upon entering the system. This stage involves cleaning the data of common occurrences within textual data that will influence the quality of any output (Honnibal & Montani, 2017). Removing punctuation, adding lexical markers (such as part of speech) needed for other tasks such as named entity recognition (Manning, et al., 2014) or splitting a large document into individual sentences to be analysed by the system separately are all examples that may occur during pre-processing.

For classification, an associated class label needs to be assigned to every document. The dataset being used may already be pre-labelled or an unsupervised algorithm like clustering may be used to produce training labels based on the resulting groupings (Károly, et al., 2018). Whilst manual labelling may allow a classifier to better generalise the relationships in the dataset, a key issue surrounding medical data specifically is the lack of readily available annotated data.

Once the text has been cleaned, the next stage is feature engineering/selection. This stage transforms the cleaned text into a series of interpretable vectors usable by a machine learning model. Feature engineering can be broken down into two distinct sub-categories: frequency-based methods such as a bag of words or TF-IDF (Aizawa, 2003), and context-based techniques like word embeddings (Mikolov, et al., 2013). Both approaches are outlined in Section 3.3. To achieve the best set of features for a task, the introduction of stop words at this stage can help prevent the models that follow from focussing on unimportant words that appear in the text.

With a vector of features representing the each original document, the information is then used to train a machine learning model. This training may be unsupervised or supervised learning depending on whether the dataset has associated labels.

The final stage in the pipeline involves evaluating the output from the earlier steps on a previously unseen portion of the dataset. The format of the output will depend on the goal of the system. For text generation, the output may form a singular word to finish a sentence or an entire answer when looking at a questioning and answering system. For text classification, the output will be the predicted label. Performance metrics such as precision, recall and *f-score* (Sokolova & Lapalme, 2009) are used to evaluate the ability of the machine learning model to understand and generalise relationships in the data presented during the training stage.

As the development of a natural language pipeline is an iterative process, insights garnered from the outputs of a model can be used to adapt stages in the pipeline to better achieve the goal. These adjustments can be made to a small elements in the pipeline: for instance adding or removing vocabulary words to a stop word list in data pre-processing, or a large element replacing the classification model.

2.4 Existing literature surrounding clinical data modelling

For many years, clinical data modelling has been a topic and has been approached using numerous ways, including rule-based, machine learning, deep learning, and hybrid systems (Fu, et al., 2020). The common trend across many publications is that the rate at which biomedical literature and electronic health records have and will continue to expand has left the ability to distil information and create hypotheses for research unmanageable (Spasic, et al., 2005). Instead, the authors discuss the idea that including a domain-specific ontology is necessary for representing the semantics of a clinical dataset. While incorporating such ontologies may benefit or be required for an information extraction task with specific biomedical indicators such as drug and protein names, this added layer of clinical expertise may not be needed for text classification.

For an approach to incorporate an ontology, that ontology must be an exhaustive dictionary of terms and each terms associated features. In the clinical domain, the ontology used is the metathesaurus provided through the Unified Medical Language System (UMLS) (Bodenreider, 2004), in which each term is mapped to a unique identifier (often abbreviated to CUI in literature). Due to the complex nature of a medical dictionary, an additional type unique indicator (TUI) is also mapped to terms to distinguish between terms with multiple meanings. In addition to a dictionary of terms, the UMLS metathesaurus also holds a series of controlled vocabularies used by health organisations worldwide. Examples of these include the ICD-10 (World Health Organization, 2004) and SNOMED clinical terms used by Digital Health and Care Wales (DHCW).

A Natural Language Processing system that leverages the UMLS Metathesaurus will need to create a dictionary lookup method or incorporate a pre-built system. Of the existing systems, the National Library of Medicine's MetaMap (Aronson, 2001) and the Mayo Clinic's clinical Text Analysis and Knowledge Extraction System (cTAKES) (Savova, et al., 2010) are the most used across literature, including the most accurate approaches to named entity recognition challenges (Uzuner, et al., 2011; Uzuner, et al., 2012). Whilst cTAKES incorporates an extended dictionary of UMLS terms and relevant synonyms to extract clinical concepts, the method relies on an exact matching approach for concept extraction. This rigidity means that unexpected acronym usage, word shortening, or spelling mistakes will cause a token to be unreadable by the system and be mislabelled. Other approaches like MaxMatcher (Zhou, et al., 2006) implement a fuzzy pattern matching technique based on the idea that it is common to have term variations within biological concepts, which reduces the accuracy of an exact matching approach. This idea has been taken further by Soldaini and Goharian (2016) with QuickUMLS to capture lexical variation that occurs within the English language. QuickUMLS provides a faster method for mapping clinical concepts from a document to its unique UMLS identifier when compared to the cTAKES exact matching approach whilst retaining a high degree of accuracy. However, the literature primarily uses cTAKES as the choice for medical named entity recognition.

Table 1 Natural language processing challenges using clinical domain data

Challenge Organiser	Challenge Task	Dataset Size
i2b2/VA (Uzuner, et al., 2007)	Anonymisation and classification of smoking patients	Training: 398 documents Testing: 104 documents
i2b2/VA (Uzuner, et al., 2011)	Concept extraction, assertion and relationship classification	Training: 394 documents Testing: 477 documents
I2b2/VA (Uzuner, et al., 2012)	Concept extraction and entity co-referencing	Training: 688 documents Testing: 454 documents
i2b2 (Sun, et al., 2013)	Temporal relationship extraction	310 documents
ShARe/CLEF eHealth (Suominen, et al., 2013)	Concept extraction	Training: 200 documents Testing: 100 documents

Due to the lack of pre-annotated clinical data availability, a lot of published research uses Natural Language Processing challenge datasets. These datasets are described in Table 1 with their organisation, challenge task and dataset size. While the datasets presented in these challenges provide an opportunity for community-based learning, they exemplify some of the issues with clinical Natural Language Processing. The challenges are highly specialised towards creating an expert system and have a small amount of data to train off of. Whilst the accuracy of systems produced due these challenges is reasonably high, there is no testing available to check how each one would perform on a non-curated dataset. In addition to this, the most accurate method found for the smoking challenge (Uzuner, et al., 2007) was to include additional outside data to train the model on (Clark, et al., 2007). This improved performance indicates that the size of the initial data was not sufficient enough to effectively achieve the classification task set out in the challenge.

Further research has been carried out using these challenge datasets in conjunction with deep learning techniques. Wu et al. (2018) implement a recurrent neural network model to improve upon the results of previous challenge entrants, training on clinical examples present within the MIMIC-III database (Johnson, et al., 2016). Similarly Lee et al. (2020) and Zhang et al. (2021) provide pre-trained models for tackling clinical named entity recognition tasks like the 2010 i2b2 challenge (Uzuner, et al., 2011) using a transformer and a recurrent neural network architecture respectively, achieving greater results than those shown by the initial challenge competitors. Zhang et al. (2018) present another deep learning method for concept extraction by combining recurrent neural networks with the attention mechanisms used in transformers. This approach was then evaluated against the existing tagging systems MetaMap, cTakes and QuickUMLS on the ShAre (Suominen, et al., 2013) challenge dataset. The approach achieves comparable accuracy to the other deep learning methods used on similar tasks while outperforming the three methods directly compared.

Whilst concept extraction or named entity recognition can be the goal of the research, it may also be used in an attempt to create a system for accurate text classification. The rest of this section provides an overview into research carried out into document classification, focusing on work carried out using healthcare data. Classification is the task of associating dataset members with a label. There are a several variables to consider when selecting an approach to a classification problem. The first involves the variation in labels available to the model, known as the supervision level. These approaches to supervision are:

- An unsupervised approach like clustering where none of the data is annotated with an associated label. This approach groups individuals into categories based solely on the data present, learning unexpected associations or relationships between data.
- A semi-supervised approach can be where the test data and a subset of the training data is annotated. Implementing a semi-supervised approach like label propagation (Zhu & Ghahramani, 2002) emphasises that similar data points will appear close together in Euclidean space. The algorithm can then begin to associate data groups together and assign labels based on the existing labels in the smaller subset. The resulting “complete” set of labels can be used as a fully annotated dataset.
- A supervised approach is where all elements of the dataset are labelled. This is the common approach to text classification but requires pre-processing of the data, sometimes in conjunction with the help of expert knowledge in the field to which the dataset belongs. This approach is again one of the reasons why some of the challenges in Table 1 are heavily used in document classification due to the lack of publicly available clinical datasets.

The number of target labels will also affect the accuracy of different classification methods. An algorithm that achieves the best accuracy in a binary classification problem (two labels) may not outperform other algorithms for a multi-class classification problem or, at times, is not suitable. Additionally, the balance of classes within the dataset also needs to be considered. Different models will perform better if each class is equally represented within the data compared to the same problem in an imbalanced dataset. Working with an imbalanced dataset in a classification problem risks overestimating how well a model has fitted the data. For instance, if a single class represents a large portion of the data and every data point gets labelled as that one class, the model can look accurate whilst it has failed to model any of the other classes present. A suitable performance metric must be chosen to evaluate the model, such as the F-Measure or Matthew’s Correlation Co-efficient (MCC).

Research produced by Tang et al. (2015) employs many methods to classify four publicly available datasets for sentiment analysis. These datasets provide a multiclass classification problem, with three of them including five labels, and the fourth having ten class labels. Whilst the research carried out does not belong specifically within the clinical domain, the models created by Tang et al. (2015) have been referenced as state of the art models in other papers. The paper compares support vector machines, convolutional neural networks, and recurrent neural networks for the task, alongside the feature generation techniques discussed in Chapter 3. The report shows that for the non-clinical classification task presented the recurrent neural network outperformed the other methods with the exception being the dataset with ten class labels in which it performed considerably worse. For feature generation, Tang et al. (2015) found that implementing a paragraph vector mechanism for creating word embeddings performed best. However, the paper does not consider vector space models available like TF-IDF as an alternative feature engineering method to word embeddings.

Inside the clinical domain, sentence and document level text classification has been implemented for many subdisciplines like patient phenotyping, drug relationships and speciality extraction. Recent research into these areas focuses on comparing traditional machine learning models to deep learning models. Rajendran and Topaloglu (2020) compare approaches to determining patients’ smoking status in both a binary classification problem and a three label multi-class problem. The research compares both the impact of the feature generation methods term frequency-inverse document frequency and word embeddings as well as different classification models. The results show that a convolutional neural network performed best in binary classification but lost to the Naïve-Bayes approach in multi-label classification. Whilst the approach achieves reasonable results, elements of the research need attention. As previously seen, clinical data has an issue with a lack of available data. The data presented in this research only belongs to 781 patients and does not state whether the data is balanced or imbalanced. The research also shows that with the recurrent neural network approach, there is an increase in the accuracy of the multi-class problem when increasing the vocabulary to include bi-grams (two-word phrases). However, this work is not echoed for the other models to test if this it could improve their accuracy.

For larger multi-label classification problems, work done by Gehrmann et al. (2018) compares the use of cTAKES to leverage concept information from the text documents to the application of a convolutional neural network architecture when phenotyping patients into ten categories. The valuable information gathered from the research is that the transformation of raw text into identified concept markers did not significantly improve the accuracy of the other used models, showing that the relationships needed for classification already exist with the text without the addition of ontologies bloating the pipeline. The research also states that performing term frequency-inverse document frequency on the data improved the performance of all the models. Research done by Krsnik et al. (2020) tries to classify different knee conditions based upon information found within radiology reports using Naïve-Bayes, logistic regression, support vector machines, random forest classifiers and convolutional neural networks. The research is completed using a small dataset size of 1,295 radiology reports with unbalanced classes. The best model presented achieves high accuracy with the three largest classes in the dataset; however, it struggled to accurately portray examples of the smaller classes.

When narrowing down the problem space of text classification to areas wherein text classification had been used on clinical specialities, work carried before this thesis by Weng et al. (2017) must be considered. The work involves producing a pipeline similar to the one described in this thesis apply traditional and deep learning models to the problem space. What was found by Weng et al. (2017) was that the traditional classifiers employing word vector representations for feature engineering (see Section 3.2) presented a better F1-Score accuracy than those achieved by deep learning methods using word embeddings. This mirrors the results shown later in this thesis (see Section 5.2.2 and Section 5.3.2).

Whilst the classification pipeline used in this thesis aligns with the methodology shown in Weng et al. (2017), there are notable differences that separate the two. The first is evaluating the feature space using feature generation and parameter selection. To improve the initial presentation of features to a model, Weng et al. (2017) have chosen to include cTAKES (Savova, et al., 2010) to extract medical concepts within the text, converting all variations of a concept to a single unique identifier. The work carried out in this thesis avoids the inclusion of such ontologies, choosing to attempt to exploit an idea that the same concept may be presented differently depending on the specialism being referred to. By relying on actual words and phrases instead of these identifiers, this approach removes the opportunity for an ontology to miss a partial phrase in a later live example that would form an out of vocabulary word.

Another key difference occurs when looking at the selection and cleaning of the datasets for the system. The work in this thesis uses all the data available for the purpose of classification whereas Weng et al. (2017) employ two curated datasets. The first is a small set of notes containing just 431 letters across six specialties. Although much larger at 542,744 notes, the second dataset encounters an issue wherein the selection process removes 83.2% of all clinical notes (reducing the number to 91,237). This occurs because the labelling technique used relies on consultant doctor’s specialism rather than the individual letter. Weng et al. (2017) removed all notes correlating to a specialist that had more than one speciality assigned to avoid mislabelling cases. The size difference is not attributed to this alone as there has also been a manual decision to limit the specialities present to the top twenty-four.

There is no guarantee that a specialist cardiologist sees a patient under the guise of a potential cardiology issue and that the actual outcome is an alternative, similar specialism. In comparison, the dataset used in this thesis for general specialism classification is labelled by the general practitioner on a per case basis which helps (but does not guarantee) to ensure a correct label is assigned to each document. The data itself and associated specialisms were also randomly selected from a larger dataset of NHS Wales documents rather than curated. As explained further in 4.2.3, the only removal of documents from the dataset involved those assigned to a speciality with a limited selection that could be considered significant or where test documents were mixed into the database. The approach taken towards data procurement only reduces the dataset size by 7.5% instead of 83.2%.

The issue with selecting just a portion of the overall dataset for research purposes is not limited to Weng et al. (2017). Table 2 presents examples of research that extract a small training and testing set compared to the overall dataset available. Cocos et al. (2017) was forced to exclude a substantial portion of their dataset due to a reliance on crowdsourcing to label the training data. However, the benefits to using a small subset of data for research purposes was not presented in the other publications listed in Table 2. For example, Cohen et al. (2016) explicitly state that maintaining both a balanced dataset and increasing the size of their training set from forty to two hundred examples increased the F1-score by ten percent. However, the authors do not try and expand their study by including other examples from their already available set of data.

Table 2 Reductions in used data by clinical domain researchers

Author	Research Goal	Total Dataset Size	Used Dataset Size
Gustafson et al. (2017)	Classifying patients for atopic Dermatitis conditions	2.5 million clinical notes 43,268 related notes	Training: 562 Testing: not disclosed
Cohen et al. (2016)	Predicting surgery candidates for paediatric epilepsy	Clinical progress notes for 6,343 patients	Training: 40 – 200 Testing: not disclosed
Cocos et al. (2017)	Crowdsourcing electronic health record labelling	10,880 unlabelled sentences	Training: 100-717
Castro et al. (2017)	Classifying breast imaging radiology reports: machine learning versus a rule-based system	2 million radiology reports	Rule-based: 1560 ML Training: 360 ML Validation: 60 ML Testing: 179
Fodeh et al. (2018)	Classifying if documents contain pain assessment indicators.	99,481 clinical notes	Training:705 Testing: 353
Weng et al. (2017)	Classifying medical subdomains	542,744 clinical notes	Training and Testing: 91,327

Other research into medical speciality classification includes Hughes et al. (2017). A convolutional neural network approach was used to classify medical text into twenty-six categories based upon pre-classified encyclopaedic articles (Merck Sharp & Dohme, 2021). Work completed found that bag of words and Word2Vec outperformed Doc2Vec when implementing the neural network. Whilst the paper states that the data was split into balanced training and testing sets for each category of four thousand and one thousand respectively, there is no mention of individual categories' performance. Instead, there is just an overall accuracy for each feature generation method. In other languages, Faris et al. (2020) implements a support vector machine method incorporating binary particle swarms to extract medical specialties from a question answering system. The research achieves accuracies in the mid-eighties for their approach, with an increase in performance over other traditional machine learning approaches on their dataset. In comparison to the data used in actual clinical letters, however, the questions in their dataset are concise with less noise to filter out.

2.5 Chapter conclusion

This chapter provides an introduction into the problem space in which this thesis sits. The premise involves augmenting the existing process of hospital referrals which are currently carried out through manual interpretation of a general practitioner’s letter by a hospital consultant. Natural language pipelines (see Section 2.3) can be built to mimic portions of this existing clinical pathway, such as classifying a set of free text documents into a series of associated speciality labels.

The last part of this chapter details insights into the existing academic approaches to using Natural Language Processing within the clinical environment. Whilst existing literature showed accurate results, it highlighted two key issues when working with medical documents. Firstly, there is a significant lack of publicly available datasets, with most academics having to rely on small challenge datasets. The second involved specialising the data, either to a specific single yes or no classification task or by pruning the data to facilitate better outputs. To avoid these drawbacks, this research ensured the use of medical datasets with a significant size that haven’t been curated to fit a specific challenge goal or to increase the chances of obtaining a better end result. The results shown in this thesis can then be a fair representation of each models’ ability to classify the data.

The following chapter outlines the first two stages of the Natural Language Processing pipeline discussed in Section 2.3, wherein documents are cleaned and transformed into machine-readable inputs for use in future machine learning models.

Chapter 3 Natural Language Feature Engineering

This chapter provides an overview of the first two stages of a Natural Language Processing pipeline (see Figure 3.1), wherein data pre-processing and feature engineering techniques are used to create a meaningful semantic representation of the data in a format that a machine learning model can use for classification.

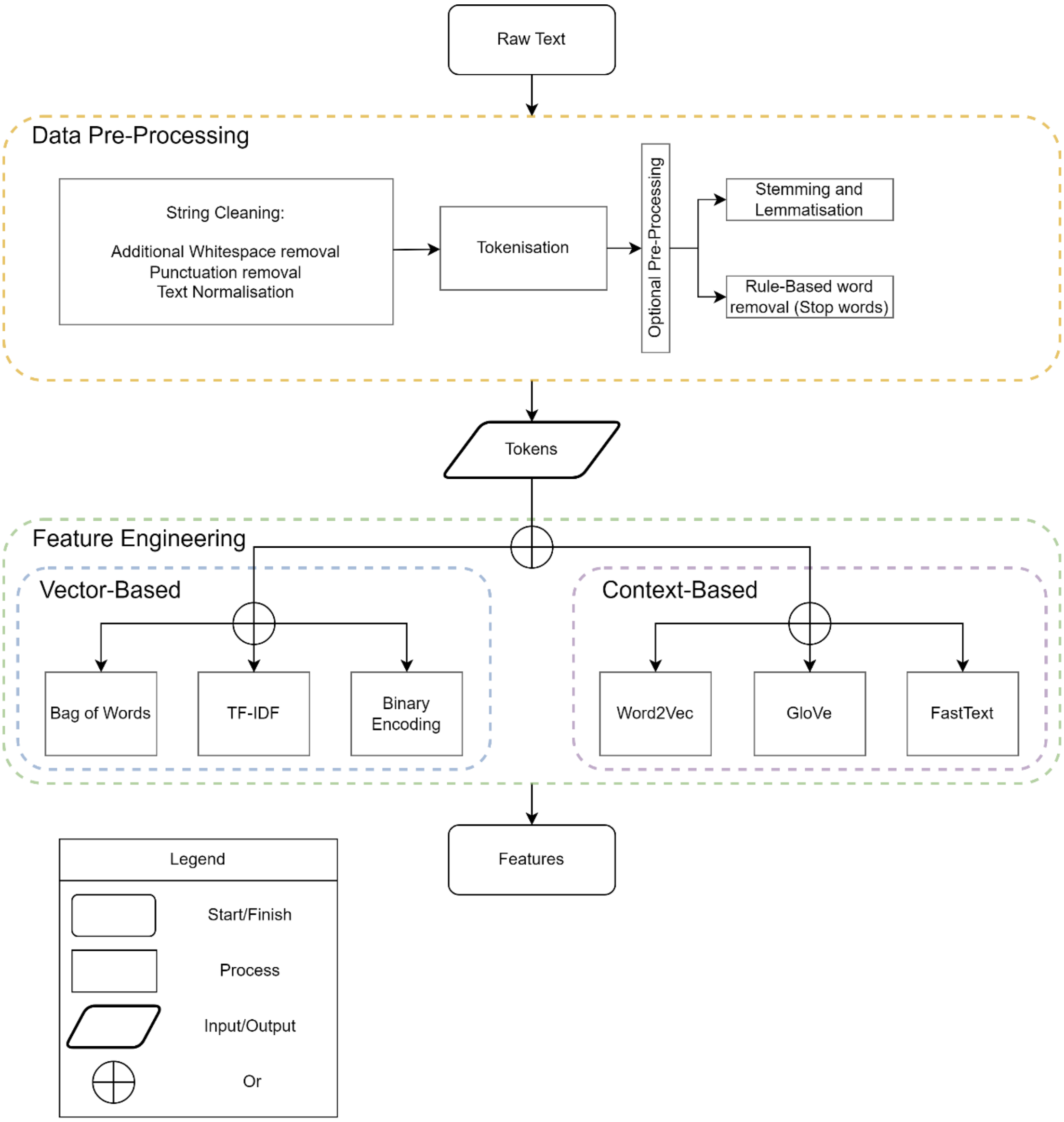


Figure 3.1 Representation of the first two stages of a classification pipeline

3.1 Initial data pre-processing

As stated in Chapter 2, raw text data is inherently noisy. The following section focuses on the first steps taken as each document enters the pipeline including processes that underpin other more complex Natural Language Processing tasks.

3.1.1 Tokenisation

Tokenisation is the process of separating a sentence or document down into individual words/tokens (Manning, et al., 2014). A tokeniser is used across Natural Language Processing tasks as most tasks are processed on a word level basis. This includes traditional machine learning models using word vectors and newer transformer-based architectures.

The approach to tokenisation was introduced into computer science by Webster and Kit (1992), wherein a token is described as the natural language equivalent of an atom (a textual element that need not be decomposed further) and the basis upon which all processes are built from.

The tokenisation process is commonly done alongside the use of regular expressions (Bird, et al., 2009) to filter out meaningless elements in the documents that consist of (but are not limited to) tabs, punctuation, capital letters and special characters. One set of such rules involves incorporating the Penn Treebank annotated corpus (Marcus, et al., 1993). Initially designed for part of speech tagging, the corpus has been used to provide rules for tokenising documents of more complex occurrences including quotes and contractions. However, due to the age of the corpus (which ran between 1989-1996), some issues may arise. For instance, the use of language has evolved, containing elements that may impact the ability for the treebank to be useful in situations with a more modern style of writing (e.g., social media). Whilst the inclusion of such rules may benefit the overall accuracy of a system, more recent pre-trained corpora of word vectors and embeddings such as GloVe (Pennington, et al., 2014)(see Section 2.3.2) split contractions directly into prefix and suffix pairs instead of expanding them. This prevents issues where two or more words share the same contraction. Whilst this is rare, it can occur when dealing with possessives such as ‘he’s’, which can be expanded into ‘he has’ or ‘he is’. There can be a significant difference in meaning between the words ‘has’ and ‘is’, especially when dealing with a dataset in a clinical scenario.

A step that may be taken at the tokenisation stage is the inclusion of a stop words list. Stop words refer to inherently noisy elements within textual data that may provide little to no benefit to any Natural Language Processing task. A stop word list can include some of the possessives discussed above, reducing the need to expand contractions. There are some default stop word dictionaries that can be downloaded for multiple languages, provided by both NLTK (Bird, et al., 2009) and SpaCy (Honnibal & Montani, 2017). However, these curated lists will need updating to fit the context area of a dataset.

The question that must be asked when tokenising is whether singular words are enough to provide an accurate depiction of the semantics of a document. In cases such as a clinical environment, conditions, tests, and treatments often span multiple words. Including a wider window for tokenisation also requires a stop word list to be updated. In the examples referenced above, words such as ‘no’ are present in the stop word list and therefore omitted from any feature engineering. However, when including phrases at the tokenisation stage, the need for these negatory terms is useful. For instance, being able to differentiate between documents containing the phrase ‘lesions present’ and ‘no lesions present’ could be imperative to deciding whether a patient would get attributed to the dermatology speciality or not.

3.1.2 Stemming and lemmatisation

Both stemming and Lemmatisation are morphological approaches to pre-processing textual data. Stemming is a process first published by Lovins (1968) in which Lovins described a 2-step, longest-matching algorithm to reduce words to their root form by removing inflectional affixes. For example, the words ‘travel’, ‘travelled’ and ‘travelling’ can all be reduced to the stem ‘travel’ whilst still retaining their meaning. Porter (1980) adapted this approach and implemented a 5-step algorithm for suffix stripping. Porter (2001) further developed this idea by introducing a programming language called Snowball, which included an updated stemming algorithm dubbed Porter2.

Issues can occur when adding stemming to a natural language pipeline due to the complexity of language. Without optimization, a stemming algorithm can run into two pitfalls. The first pitfall that occurs is over stemming. The stemming algorithm is overzealous, reducing different words with different meanings to the same root word or the root word’s meaning is different to the original. For example, the Porter stemmer reduces ‘meaning’ to ‘mean’. While the original word is a participle used to explain a concept, the derived root form can mean the same, hurtful to someone or a mathematical way of averaging. The other pitfall, under stemming, describes when the algorithm does not find the same root stem for two inflexions of the same word (an inflexion being a change in word form such as tense). This issue arises more often in Romance languages such as French where articles differ based on the subject’s gender. There are further issues in languages such as Welsh and German where the language includes multipart words (Dale, et al., 2000).

To alleviate the issues that arise with stemming, the process of lemmatisation does not focus purely on removing suffixes from words. Instead, the process transforms words into their lemma which may include adding or replacing the word entirely as shown below:

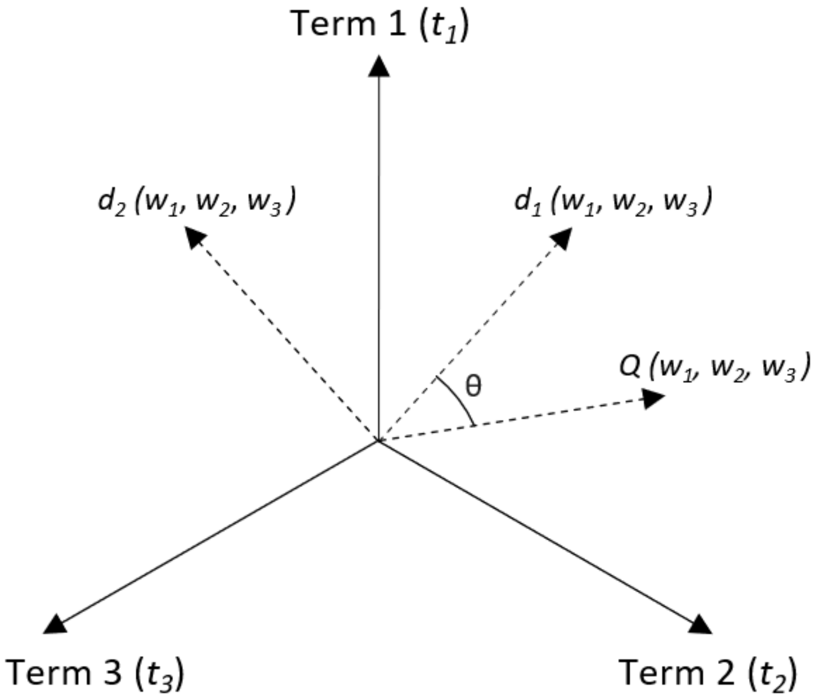
Stem - Worse -> wors
Lemma - Worse -> bad

The lemmatisation process relies on additional external sources to complete these more complex transformations. At minimum, two corpora are required: a dictionary of all words and a corpus of stop words. A more complex lemmatiser will also require the inclusion of a part of speech tagger to differentiate between verbs and nouns, akin to the example shown in Section 2.1 with the homonym ‘leaves’ having two meanings in different contexts. The lemmatisation process works by scanning the corpora for a matching word and replacing it when appropriate. However, employing a lemmatiser requires a substantial increase in both processing power and time compared to the stemming process described above.

The inclusion of either transformation into a Natural Language Processing pipeline is to try and improve the generalisation of a model by reducing the vocabulary and the model’s ability to overfit to individual words present within a dataset. Whilst the above statement is true, there are scenarios in which the inclusion of either transformation would be detrimental to the accuracy of a model. The first is the context area of the dataset, a scenario where a series of non-standard words may get reduced and lose their semantic meaning. The second relies on the methods used for feature engineering that are discussed later in this chapter. A lemmatised vocabulary will change the associated frequencies for features in a bag of words vector representation. This effect would be compounded further when using a different vectorisation approach such as term frequency-inverse document frequency. The weight of each word changes with the number of times it occurs within the document set.

3.2 Feature engineering: frequency-based vector representation

As previously stated, there are two distinct methods for feature generation. The first is frequency-based word vector representation to create vector space models. This section provides an overview of the methods behind vector space models before detailing the two encoding methods used for experiments later in this thesis.



Term 1 (t_1)
d_2 (w_1, w_2, w_3)
d_1 (w_1, w_2, w_3)
θ
Q (w_1, w_2, w_3)
Term 3 (t_3)
Term 2 (t_2)

Figure 3.2 Vector space model for three documents with a vocabulary size of three

For a vector space model, the goal is to create a representation of each document (*d*) in n-dimensional feature space (Salton, et al., 1975). To accomplish this, first, a vocabulary of features (both words and phrases) is extracted from the dataset's entirety. A document is then transformed into a 1-dimensional vector of feature weights using an encoding method. Iterating this process across the entire dataset creates a sparse, semantic matrix with the dimensions of D x V (the set of all documents x the total set of vocabulary features).

After encoding, each document can then be expressed as a vector such that *d* = (w_1, $w_2 \ldots w_n$), wherein *w* is the encoded weight for each term in the vocabulary (Lee, et al., 1997). Figure 3.2 provides an example of a vector space model for a dataset that consists of three documents (d_1, d_2, *Q*) and a vocabulary of three terms (t_1, t_2, t_3). As shown, the combined weighted value of each of the presented terms determines the final position of a document within the vector space.

Creating a vector space model allows for the similarity between two documents to be measured. An example of such a measure is the cosine similarity shown Figure 3.2 where the distance between two vectors is equal to the cosine (θ) of the angle between them.

$$similarity\ (Q, d) = \cos\theta = \frac{\vec{Q}\cdot\vec{d}}{||\vec{Q}||\ ||\vec{d}||} = \frac{\sum_{i=1}^{n} w_{q,t} w_{d,t}}{\sqrt{\sum_{i=1}^{n} w_{q,t}^2}\sqrt{\sum_{j=1}^{n} w_{d,t}^2}} \qquad (1)$$

where:

- Q is the new query document
- d is an existing document for comparison
- $\frac{\vec{Q}\cdot\vec{d}}{||\vec{Q}||\ ||\vec{d}||}$ is the dot product of the two different document vectors (Gudivada & Rao, 2018)

To create these comparable unit vectors, the initial document vectors require normalisation to reduce the bias towards documents of longer lengths. This normalisation factor is the denominator in (1) and is explained by:

$$\sqrt{w_1^2 + w_2^2 + \cdots + w_n^2} \qquad (2)$$

where:

- w is vocabulary term within a document vector
- *n* is equal to the length of the document vector (Bagga & Baldwin, 1998)

With the creation of these structured document vectors, machine learning models used for document classification can interpret and evaluate the contents of a document and classify individuals into labelled sets.

The method chosen for encoding the vocabulary within a document set will dramatically vary values for individual feature weights. Whilst the binary encoding method (dummy encoding (Boole, 1854)) is noted as a precursor to other vector space model encoding methods, the bag of words and TF-IDF methods discussed in Section 3.2.1 and Section 3.2.2 respectively.

3.2.1 Bag of features

The *Bag of Features* approach to feature selection employs a strategy based upon an idea set out by Harris (1954). The theory states that the distributional structure of any document can be represented by the frequency of terms relative to the other terms present within the document and is complete without the inclusion of other added rules. When applying this approach to textual data, a “Bag of Words” approach means creating a vector containing frequency counts of words present across a dataset (Guyon & Elisseeff, 2003). The approach has been extended outside of textual information to include visual object recognition tasks (Li, et al., 2010; Sarwar, et al., 2019; Zhang, et al., 2010) by either labelling sections of the images or extracting combinations of letters and numbers. The rest of this section describes an example of Bag of Words in action.

Taking the following three letters as the entire dataset, the two steps taken by the approach are as follows.

Letter 1 - The man does not have arthritis.

Letter 2 - The man had arthritis.

Letter 3 - The man has been congested.

As mentioned in Section 3.2, the first step iterates over each document present within a dataset and creates a vocabulary of terms consisting of words and/or phrases. There are ten unique occurrences of terms within the above examples when looking at words alone. The next step then creates a vector for each of the documents in the dataset and assigns frequency values to each term within the vocabulary as shown in Table 3. The resulting fixed length vectors are understandable by a machine learning model to be used for clustering or classification purposes.

Table 3 Bag of Words feature table

	Letter 1	Letter 2	Letter 3
The	1	1	1
man	1	1	1
does	1	0	0
not	1	0	0
have	1	0	0
arthritis	1	1	0
had	0	1	0
has	0	0	1
been	0	0	1
congested	0	0	1

From the information shown in Table 3, some of the issues of using the Bag of Words approach can be seen. Increasing the size of the dataset from two to three documents increased the vocabulary size by thirty percent on single words alone. The inclusion of phrases would exponentially increase the dimensionality of the vectors created. However, without including phrases a level of context is lost. A task requiring an elevated level of granularity may not be able to differentiate between a vector containing a term and a negated term as is often found within medical letters. The final issue arises with the approach's method regarding each term as equally important. From the example in Table 3, the Bag of Words approach considers each letter inherently similar because they all share the same two starting terms.

3.2.2 Term Frequency–Inverse Document Frequency

Term Frequency-Inverse Document Frequency (TF-IDF) is an alternate approach to vector space model creation that can be used to counteract the issue that arises within a Bag of Words model about term importance. The method builds upon the idea of term specificity first described in Spark Jones (1972) to determine the semantic value of any present term in a document. The idea states that a highly generalised term is less likely to be an influencing factor when measuring the representation of an overall document (Aizawa, 2003). The formula used to create a TF-IDF vector representation of a document can be seen in (3).

$$tfidf(t, d\ , D) = tf(t\ , d) \times idf(t, D) \tag{3}$$

where:

- t is a vocabulary term
- d is a single document
- D is the set of all documents
- *tf* is the term frequency - *tf(*t, d) = $\frac{f_d(t)}{|d|}$
- $f_d(t)$ is the frequency of term *t* in document *d*
- *idf* is the inverse document frequency - $idf(t, D) =$ $\log\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right)$

Simply put, the TF-IDF values are a product of the featured terms within a document in correlation to the number of appearances of that term across the entire dataset. The term frequency part of the formula deals with the terms solely inside the document analysed. It ranks the unique terms (t) by occurrence over the number of words present within the document. Whilst the examples used in Table 4 all use a single occurrence of each of the terms, it shows the difference in the frequency values dependent on the length of the given document.

Table 4 Term frequency table as part of TF-IDF

	Letter 1	Letter 2	Letter 3
The	1/6	1/4	1/5
man	1/6	1/4	1/5
does	1/6	0/4	0/6
not	1/6	0/4	0/6
have	1/6	0/4	0/6
arthritis	1/6	1/4	0/6
had	0/6	1/4	0/6
has	0/6	0/4	1/5
been	0/6	0/4	1/5
congested	0	0	1/5

Table 5 shows the impact of adding an extra "the" to Letter 3 and that without the second half of the TF-IDF formula a non-descript article would end up as the most important feature.

Table 5 Term frequency table for updated letter 3

	the	man	does	not	have	arthritis	had	has	been	congested
Letter 3	2/6	1/6	0	0	0	0	0	1/6	1/6	1/6

The inverse document frequency part of the formula transforms the term frequency counts into representations of term relevancy. The weight of each term present within a document is calculated by taking the log of the number of documents in a dataset divided by the number of documents containing the term.

Table 6 outlines the calculations made to evaluate the inverse document frequency for each of the ten terms present in the example vocabulary. With this information, it can be seen that when dealing with this half of the formula, adding a duplicate term to a document would not change the outcome as the presence of each vocabulary term is denoted as a binary value.

Table 6 Inverse document frequency values for the example data

	Letter 1	Letter 2	Letter 3	IDF Value
The	1	1	1	Log(3/3) = 0
Man	1	1	1	Log(3/3) = 0
Does	1	0	0	Log(3/1) = 0.477
Not	1	0	0	Log(3/1) = 0.477
Have	1	0	0	Log(3/1) = 0.477
Arthritis	1	1	0	Log(3/2) = 0.176
Had	0	1	0	Log(3/1) = 0.477
Has	0	0	1	Log(3/1) = 0.477
been	0	0	1	Log(3/1) = 0.477
congested	0	0	1	Log(3/1) = 0.477

Combining the two halves of the formula results in the matrix shown in Table 7. Common terms across all three documents are seen as unimportant, whilst those that differentiate between the documents are weighted higher. With such a small dataset, the words that appear in only one document are significantly higher than those that appear in two. With a much larger dataset like the ones discussed in Chapter 4, the importance of more common grammatical terms used to connect two ideas (i.e., “does” and “not”) is subdued.

Table 7 Final TF-IDF matrix for the example dataset set out in Section 2.2.1

	Letter 1		Letter 2		Letter 3	
	Formula	Value	Formula	Value	Formula	Value
the	1/6 * 0	0	1/4 * 0	0	1/6 * 0	0
man	1/6 * 0	0	1/4 * 0	0	1/6 * 0	0
does	1/6 * 0.477	0.0795	0 * 0.477	0	0 * 0.477	0
not	1/6 * 0.477	0.0795	0 * 0.477	0	0 * 0.477	0
have	1/6 * 0.477	0.0795	0 * 0.477	0	0 * 0.477	0
arthritis	1/6 * 0.176	0.294	1/4 * 0.176	0.044	0 * 0.176	0
had	0 * 0.477	0	1/4 * 0.477	0.119	0 * 0.477	0
has	0 * 0.477	0	0 * 0.477	0	1/6 * 0.477	0.80
been	0 * 0.477	0	0 * 0.477	0	1/6 * 0.477	0.80
congested	0 * 0.477	0	0 * 0.477	0	1/6 * 0.477	0.80

3.2.3 Observations

Vector space models for document representations provide an understandable method for presenting structured data to a machine learning model from a series of unstructured text documents. While collecting the frequency of terms across a dataset is simplistic compared to later techniques in Section 3.3, the methods presented in this section have shown to produce accurate results (Tang, et al., 2015). When comparing bag of words and term frequency-inverse document frequency, the first is easier to interpret when looking at the matrix and how values are generated but the latter produces a better result on machine learning tasks (Weng, et al., 2017). Work carried out by Fan and Zhang (2018) shows that the model chosen for classification will dictate which feature generation is best and also states that the inclusion of phrases (bi-grams and tri-grams) had a greater influence on the machine learning model's ability to classify. However, Fan and Zhang (2018) did not compare the ability of term frequency-inverse document frequency to the bag of words when using feature sets that included phrases.

It is important to note that implementing a stop word list as described in Section 3.1.1 will significantly change the number of variables stored in the matrix which are produced due to one of the vector space models described in this section. Using the standardised set of stop words from the Natural Language Tool Kit (2009) Python library will result in a matrix containing only the terms "the", "man", "arthritis", and "congested". However, the formula for calculating the term frequency-inverse document frequency will still take the removed terms into account when determining the length of each document.

An issue when using vector space models is that the value assigned for each word/phrase does not consider the context of a document. Without this extra level of context, terms outside of the current word/phrase window are treated as individual occurrences that have no relevance to each other. The approaches outlined in the next section of this thesis describe an alternative feature selection method that aims to counteract this issue by defining a term as a collection of its surroundings, with the idea being to discover the meaning of a word rather than just its occurrence.

3.3 Feature engineering: word embeddings

Whilst the vector space models discussed in Section 3.2 focussed on frequency-based methods for representing words, word embeddings rely on the belief that terms can be predicted from contextual markers present in the surroundings in which they exist (Baroni, et al., 2014), expanding on the idea presented in Firth (1957). These relationships can be built in different ways, with the primary methods involving neural networks (see Section 3.3.1) or co-occurrence matrices (see Section 3.3.2) with stochastic gradient descent to train the final embedding matrix.

Creating an embedding matrix for use in a machine learning model follows a different structure to a vector space model. Whilst both methods create a vocabulary of terms, the second dimension of an embedding matrix is an unlabelled series of trainable parameters of a predefined length. The length of the dimension will depend on the relevant dataset, with a higher dimension space creating a more granular set of relationships at the cost of training time and processing power. By approaching feature generation in this way, the resulting embedding matrix should be significantly smaller than one created when using a vector space model approach whilst keeping a valid representation of the terms within the dataset rather than on a per document basis. As a result of keeping dimensions low, the approach aims to avoid the “curse of dimensionality”, wherein Euclidian distances between terms become less meaningful as the dimensionality increases (Bengio, et al., 2003). This occurs when multiple dimensions are equal between documents, as will happen when dealing with a sparse matrix like the vector space models discussed in Section 3.2.

Word embeddings are an unsupervised method of feature generation the relationships used as a dimension are less explainable than the vector space models. There is no direct, traceable correlation between the inputted words and the outputted embeddings like there is with the frequency-based representations.

To explain the goal of forming relationships through word embeddings, an example has been created in Figure 3.3. The example employs the medical specialties “Andrology” and “Gynaecology” alongside the terms for gender “Man” and “Woman”. Whilst the two speciality terms may not occur in the same letter, the idea is that similar words will appear around them.

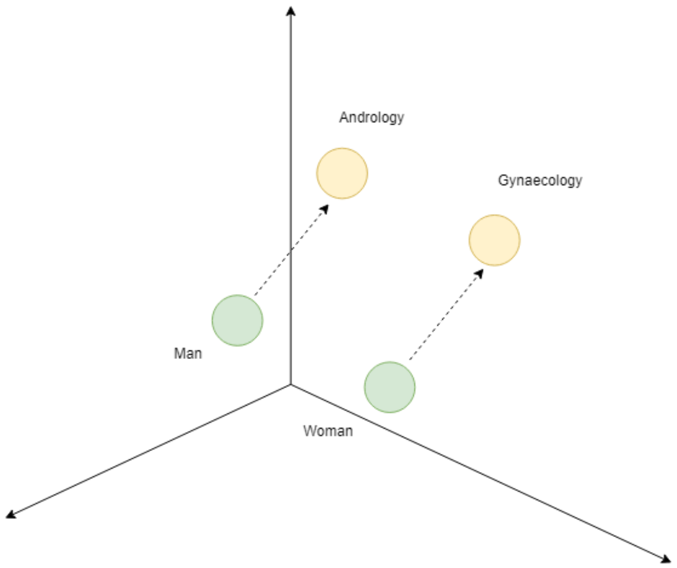


Figure 3.3 Andrology and Gynaecology word embedding gender relationship

The equation shown in (4) explains the concept. As both specialties have dealings with the reproductive system, taking the vector ([term]) of a term and switching the genders should move the vector in n-dimensional space to be in close to the equivalent function/event for the opposite gender. The following sections overview the different methods used to create word embeddings as an input into a machine learning model.

$$[\text{Andrology}] - [\text{Man}] + [\text{Woman}] = [\text{Gynaecology}] \qquad (4)$$

where:

[Andrology] is an embedding vector for the initial medical speciality

[Man] and [Woman] represent relationships used to shift the position in n-dimensional space

[Gynaecology] is the anticipated embedding vector after adjusting for the relationships.

3.3.1 Word2Vec

The Word2Vec models for creating word embeddings build upon the neural probabilistic language model outlined in Bengio et al. (2003). Introduced in 2013 by Mikolov et al., the Word2Vec architectures employ a feed-forward, back-propagation neural network to learn context based representations without supervision. The two architectures presented are a continuous bag of words and the skip-gram model. Both models are built upon the same base concept shown below, with the differences between the two models discussed later in this section.

Akin to the methods discussed earlier in this thesis, the first stage is to create a vocabulary of every term present within the dataset (*V*). Secondly, the size of the trainable embeddings must be chosen (*N*). These two elements form the dimensions of the embedding matrices used and produced at the end of training. The values within these matrices are then randomised before training begins.

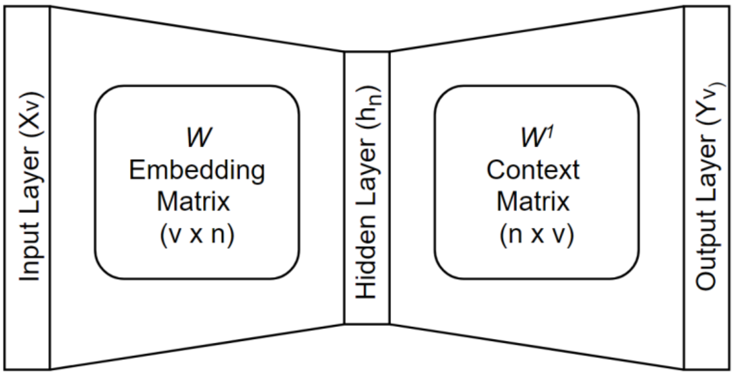


Figure 3.4 Abstracted Word2Vec continuous bag of words model

To help explain the workings of Word2Vec, Figure 3.4 outlines a continuous bag of words model that incorporates only a single context word meaning a single input term and an expected target term. The illustrated neural network is fully-connected with each node connected to each node in the next layer. Each element of the figure can be described as follows:

Input layer: A one-hot encoded vector for a single term with a length of V. As the vector uses one-hot encoding, from x_1 to x_v there will only be a single unit activated (1) and the other units will be deactivated (0).

Embedding matrix (W): A weight matrix of size V x N wherein each row is the N-dimensional vector representation of a vocabulary term from the input vector.

Hidden/Projected layer: An n-dimensional representation of the weights from the embedding matrix for the activated unit from the input vector. The hidden layer has only a linear activation function, providing no additional calculation to the received weight vector and is essentially just a copy. Alternatively, the function of this layer can be described as a lookup table.

Context matrix (W^1): Secondary weight matrix of size V x N wherein each row of the N-dimensional vector represents of a term present within the vocabulary. The vector from the hidden layer is compared to each of the vectors within this matrix and a score is computed for each.

Output layer: Outputs the probabilities for each term across the whole vocabulary of terms. The term with the highest probability is selected as the model's expected response. The softmax activation function is typically used to produce this list of probabilities from the context matrix scores.

For training, the resultant response chosen by the model will then be compared to the actual target. The weights will then be updated via backpropagation in both the context and embedding matrix to increase the probability of observing the actual output target. Explanations of the mathematics used for the training objectives and loss functions can be found in Rong (2016). During training Word2Vec employs negative sampling to reduce overfitting to common terms. Negative sampling means that whilst the weights associated with the correct outcome are always updated each epoch, only a small subset of negative samples are updated. This decrease has the added benefit of also reducing the time and cost of training the network. The context matrix is discarded and the resultant embedding matrix is used as the basis for a machine learning task after training.

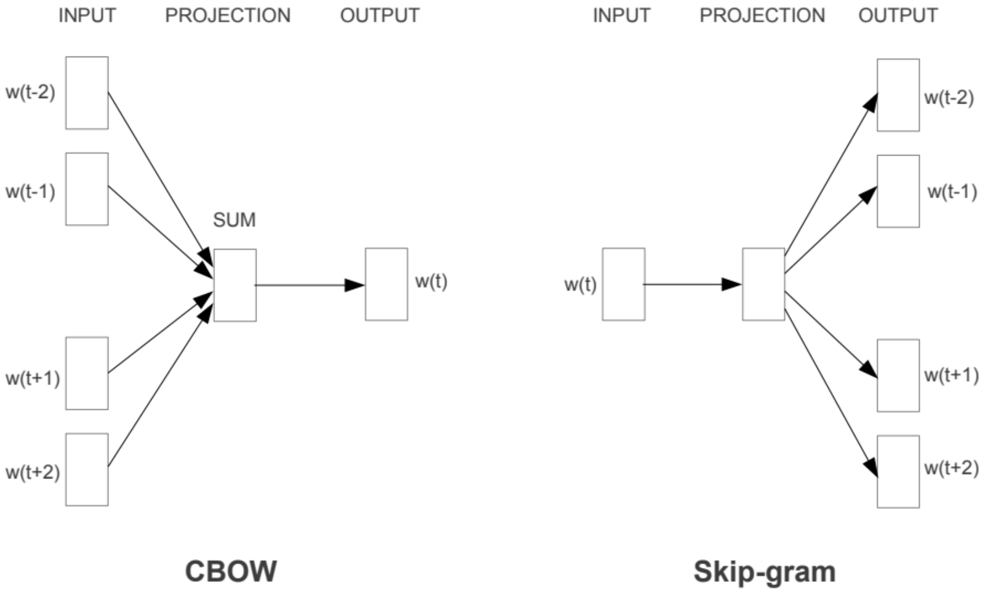


Figure 3.5 Continuous bag of words and skip-gram models for embedding matrix creation (Mikolov, et al., 2013)

The actual implementation of the continuous bag of words model (Figure 3.4) expands on
Figure 3.4 to work with a multi-word input window. Elements on either side of the word within the text are used as the context window to predict a target word. The weights of each term are gathered from the embedding matrix and the summation of the weights creates the n-dimensional representation within the hidden layer. Training the model occurs in the same way, except that the embedding matrix for every word used in the context window gets updated. The skip-gram model is the mirror of the continuous bag of words model. It relies on a single word input vector but tries to predict the surrounding context window from that singular word.
The difference in the two approaches leads to embedding matrices that excel differently. The continuous bag of words approach will better learn relationships between morphologies of the same word such as pluralisation's. The skip-gram produces a better understanding of a datasets semantic relationships. Words that may have no direct relationship such as "car" and "bus" will be closer when using this model as their predicted contexts would be similar. The best choice of model will depend on the converted dataset, with skip-gram working best with datasets that rely on a vocabulary consisting of terms that rarely appear. Using the continuous bag of words approach in this situation may result in the smothering of those rare words by frequent words that appear in similar contexts as they are used as part of an input context window multiple times. However, a significant increase in training time and processing power is required to create the skip-gram model (Mikolov, et al., 2013).

3.3.1.1 Doc2Vec

Doc2Vec extends the Word2Vec algorithm used to calculate relationships between documents instead of individual words. Le & Mikolov (2014) explain that the Doc2Vec algorithms learn from fixed length paragraph vectors created from variable lengths of text. Again, two methods are presented: A distributed bag of words and distributed memory, shown in Figure 3.6.

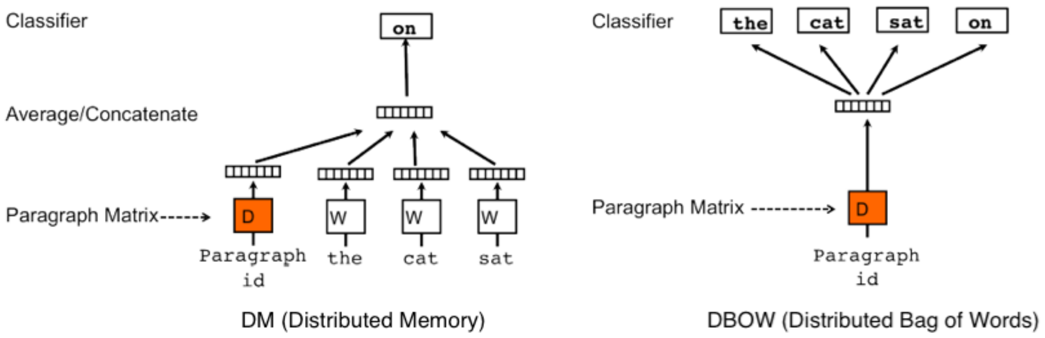


Figure 3.6 The two Doc2Vec models (Le & Mikolov, 2014)

The training method employs the same neural network approach as the previously discussed Word2Vec algorithm. By default, the distributed bag of words approach focusses solely on each paragraph matrix, forgoing updating any word embeddings during training. However, the inclusion of the Word2Vec skip-gram model can be used to train a word embedding matrix simultaneously to potentially aid performance. Conversely, the distributed memory approach treats the paragraph as an additional label, updating both the paragraph embedding matrix and the word embedding matrix at the same time. Earlier work has shown that the distributed bag of words approach outperforms the distributed memory and both Word2Vec methods when looking at a multiclass classification problem (Lau & Baldwin, 2016). However as stated in Le & Mikolov (2014) there is the potential to merge the two approaches to create a more robust set of word embeddings.

3.3.2 GloVe relies on counting occurrences of words within

Introduced in Pennington et al. (2014), the GloVe algorithm learns word relations through the implementation of a co-occurrence matrix. This matrix counts the number of occurrences in which a vocabulary (*row*) word is found in a sentence alongside a context (*column*) word instead of the neural network based approach discussed earlier in Section 2.3.1. As the name suggests, the Global vectors capture relationships by analysing a word in the context of an entire dataset whilst the earlier methods captured relationships between words in a local context window.

The following outlines the stages taken to create a usable set of word embeddings with the GloVe algorithm. After the usual data pre-processing stages, a large co-occurrence matrix is created with dimensions of the vocabulary by the context. Whilst these dimensions can be equal, the context dimensions can be larger than the vocabulary. This occurs when the top X number of features are chosen to create a vocabulary, but all words are used for the context. To chart the occurrences of a word in a context a window is used like the earlier Word2Vec models. This window can be symmetric (taking tokens from either side of the word) or asymmetric (only taking tokens before the word).

$$P_{ij} = P(j|i) = \frac{X_{ij}}{X_i} \tag{5}$$

Where:

P_{ij} is the probability of word *j* will occur in context *i*

X_{ij} is the number of times *j* appears in context *i*

$$X_i = \sum_k X_{ik} \tag{6}$$

X_i is the sum of occurrences for any word in context *i*

Pennington et al. (2014) explains that the asymmetric approach works best on syntactic tasks, but the window symmetry does not significantly impact semantic tasks. Following the creation of the co-occurrence matrix, relationship extraction can be carried out by comparing co-occurrence probabilities. The probability of any word existing within a context is calculated as per (5) and (6).

The similarities of two words can then be compared using the ratio of probabilities in the context of a third word. Using the same example specialties from Section 3.3, Table 8 shows that within certain contexts that the words andrology and gynaecology can be seen as similar whilst having features contexts that keep them separated. The contexts in which one word is heavily present will result in a ratio far from equivalent, while a context where both words or neither word are present will be of close equivalency. A context where both elements are present will be slightly higher than one, and a context where neither elements are present will be somewhat lower than one.

Table 8 Andrology and Gynaecology probability occurrence and Ratio of words

	k = *male*	k = *female*	k = *fertility*	k = *nose*
$P(k\|andrology)$	High	Low	High	Low
$P(k\|gynaecology)$	Low	High	High	Low
$\frac{P(k\|andrology)}{P(k\|gynaecology)}$	>1	< 1	~1	~1

Attempting to create embeddings from a matrix the size of the initial co-occurrence matrix would require too many resources. Instead, several features are chosen to capture a high variance representation of the initial matrix. Whilst the correct number of features will vary per dataset, Pennington et al. (2014) states that there are diminishing returns above 200 dimensions.

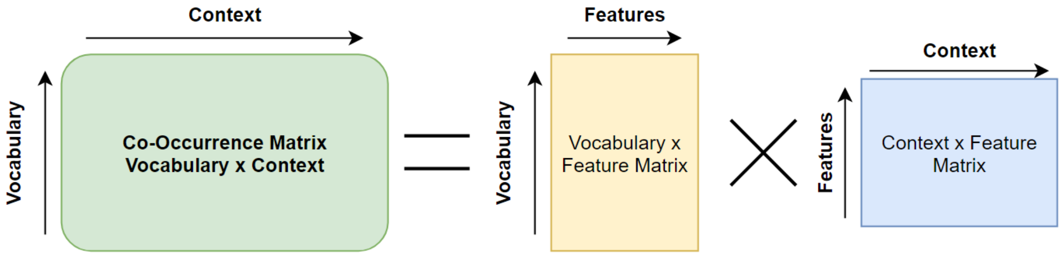


Figure 3.7 Factorization of the co-occurrence matrix into two matrices of features

As shown in Figure 3.7, two matrices are created using the vocabulary, context and new feature dimensions. Each new matrix is filled with a random set of weights for each feature and the algorithm tries to recreate the initial matrix by factorization. The model trains using stochastic gradient descent to achieve a weighted least-squares objective function, adjusting the weights in each matrix until no further progress is made. The features created during this process are again learnt rather than selected, and as there is no human control over what the model decides to select as a feature there can be no certainty as to what a feature represents.

Although there is a higher initial memory cost when using the GloVe algorithm, it is faster to train and more easily scalable than the Word2Vec models. The impact of hyperparameter initialisations (learning rate) is greater when dealing with GloVe and can affect the quality of the final embeddings.

3.3.3 FastText

The final word embedding algorithm discussed is the FastText algorithm (Bojanowski, et al., 2017). The FastText algorithm builds upon the idea of a factored neural language model (Alexandrescu & Kirchhoff, 2006) that includes both words and sub features combined with the skip-gram model discussed in Section 3.3.1. The premise works off the idea that splitting a word into component character n-grams can better represent rarer words that may appear in a dataset whilst also capturing morphisms of the same word that appear in the text.

As previously stated, FastText works by splitting a model into component features. This means that a pair of terms like "monarchy" and "anarchy" that may not be captured as similar in a different scenario will have some semblance of a relationship due to a suffix of "chy". Whilst the effect of these added affix relationships is minor when dealing with the English language, it can be helpful for languages that have compound words such as German. FastText does have its own drawbacks. As each word is taken individually and split, the focus of the training process is on a per word basis. Datasets that rely on features that occur together as a phrase may lose those connections.

Results have shown that the approach outperforms other word embedding algorithms when performing tasks such as sentiment analysis (Joulin, et al., 2017) and question answering (Mikolov, et al., 2018). While the results show that FastText should outperform GloVe embeddings on these tasks, the relative performance to other methods such as TF-IDF and bag of words is comparable. The method with the highest accuracy will depend on the dataset analysed.

3.3.4 Observations

Word embeddings provide a viable alternative approach to extracting features from a dataset to those discussed in Section 3.2. The algorithms are used to allow for more intricate relationships to be created across a dataset and help deal with situations where morphemes are prevalent. Both the neural network (Word2Vec/FastText) and co-occurrence matrix (GloVe) approach create embeddings that are similarly accurate to one another. Both approaches should be considered as options when converting a series of documents into usable embeddings before input into a machine learning model for more complex tasks. The ability to provide this context-based overview of the text comes at a greater computation cost than the vector representations. Suppose the processing power and time are a constraint when considering a method for creating a new set of word embeddings. In that case, FastText should be used over the other word embedding algorithms as it is significantly faster.

3.4 Feature reduction methods

Word embeddings are one way of dealing with the curse of dimensionality. The other way is to reduce the number of features present within the vocabulary itself. Feature reduction methods can be used as pre-processing steps when using techniques like principal component analysis (PCA) and singular value decomposition (SVD), post-processing steps when using recursive feature elimination (RFE), or as the complete pipeline for topic modelling when using latent Dirichlet allocation (LDA) (Spasic & Lovis, 2020).

The goal of implementing a feature reduction technique is to remove redundant features present within the dataset, specifically the vocabulary when dealing with text data. TF-IDF (see Section 3.2.2) vectorisation already shows a filter-based approach to feature reduction. If a feature appears too often across the dataset or too little to have any impact, it is filtered out of the final vocabulary. PCA is a matrix decomposition technique for feature reduction that transforms the data into linear combinations of existing features referred to as a principal components (Abdi & Williams, 2010). The goal of applying PCA is to discover the number of these principal components needed to represent a significant level of variance within a dataset. The first principal component is the result of a linear function that separates the original data in a way that maximises variance within the dataset. Any further principal components are computed at an orthogonal direction to any principal components that already exist. As such, the maximum number of principal components that can be assigned to a dataset is the number of features. Following this principal, each additional principal component added will represent less individual variance in a dataset. Figure 3.8 explains how the addition of each principal component produces less individual variance and has less impact on the total cumulative variance discovered within a dataset.

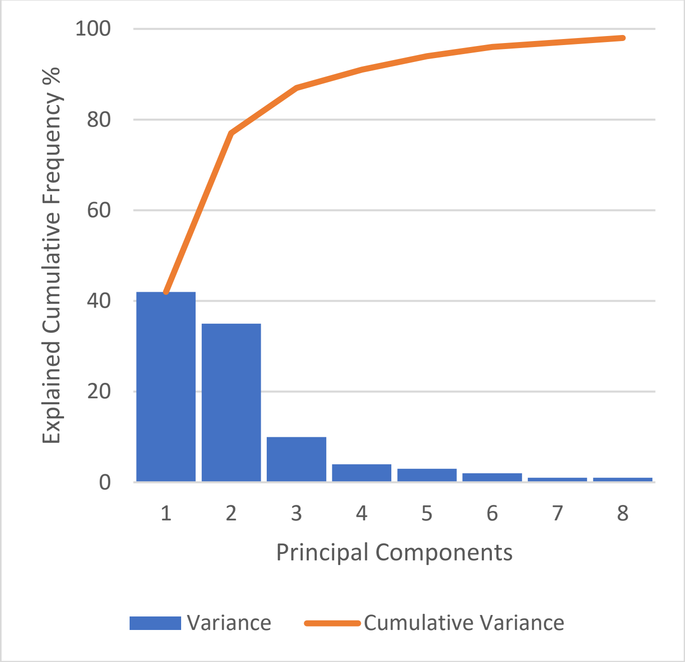


Figure 3.8 The difference between individual and cumulative explained variance using PCA matrix decomposition

SVD is another form of matrix factorisation using eigenvectors like PCA. The goal of implementing SVD is to find the best-fitting subspace for *k* dimensions, minimising the sum of squares error created between the perpendicular distance of points within the dataset to the created subspace (Blum, et al., 2013). Unlike PCA however, the SVD approach does not transform the data into independent components. The outputted matrices from applying SVD to a dataset still represent each individual feature present within the dataset. This means, if required, the outputted SVD matrices could then be used as an input for PCA.

Both methods of feature reduction are heavily used in conjunction with machine learning models. Cao et al. (2003) explains the effects that PCA has on achieving a better normalised mean squared error when employing support vector machine for time series forecasting problems. Beam et al. (2020) implements PCA and SVD as alternative methods to GloVe and Word2Vec for creating word embedding models to be used with machine learning models. When applying feature reduction directly to text categorisation, Uğuz (2011) implemented a two-stage method of feature selection to prepare the data for clustering and classification. *Ibid*. found that the performance of the models increased with a set of features that was significantly smaller than the original vocabulary.

3.5 Chapter conclusion

Presenting the data to a machine learning system in an acceptable format is as important as the model chosen for classification. This chapter has shown that there are multiple options, with salient reasons for testing each of them out.

Using a frequency-based encoding such as the bag of words or TF-IDF is helpful when creating a fast baseline model on an unexplored dataset. This initial input matrix can then be compared to the more complex models using neural networks to determine if the added computational requirement was necessary. Frequency based encoding methods also benefit from explainable white box models in comparison to black box word embedding approaches. The methods used for weighting each feature within a dataset are readily traceable to their origin and are explainable. In contrast, reducing features through techniques such as Word2Vec result in problematic relationships to examine.

Whilst there are pre-trained vectors of word embeddings available for GloVe (Pennington, et al., 2014) and FastText (Grave, et al., 2018) that consist of several billion tokens, the relationships found within a highly domain-specific dataset (such as medicine) may not be present in a set of vectors learnt from a source like Wikipedia. Combining this possibility with a similar issue in that a project may involve a small dataset may render some word embedding techniques inaccurate if an embedding matrix needs to be created without an existing, larger vocabulary.

The benefit of word embeddings is that the matrices created already exist in a dense format. This is the expected matrix format for the neural network machine learning libraries Tensorflow and Keras. This means that no transformation from a sparse matrix to a dense matrix is necessary. Comparatively, using a vector space model that requires matrix transformation may be impossible due to memory issues if the initial matrix is too large.

With more recent developments in Natural Language Processing tasks, pre-processing the data into a matrix may be unnecessary. Whilst the initial cleaning, stemming and lemmatisation still occur, transformer models take the input as a set of words and calculate relationships (feature weights) as the model trains. The following chapter introduces the modelling techniques used in this research alongside the hardware and software requirements for the experiments.

Chapter 4 Experiment Setup

Chapter 4 outlined the three datasets that are evaluated in this thesis, with each presenting different complexities such as the size of the dataset used with the general practitioner’s referral letters. In order to accomplish research question three, any approach to classification needs to be applicable with hardware constraints. As such, experiments were conducted using three consumer hardware setups. The initial anonymisation and vectorisation of patient data was carried out on an encrypted DHCW laptop. The clustering and classification methods used to produce the results in Chapter 6 were conducted using the hardware setups listed in Table 9. The first setup was used for the presenting complaints data and initial exploration into the urology dataset. The second was used for the final classification tasks with the urology dataset, as well as all tasks related to the general referral dataset. Both sets of hardware shown in Table 9 used the PyCharm Professional Edition IDE running Python 3.7 on Windows 10.

Table 9 Hardware used for clustering and classification

CPU	GPU	DDR4 RAM
AMD Ryzen 7 1700x	NVIDIA GeForce GTX 1060 6GB	16GB 2133mhz
AMD Ryzen 9 5900x	NVIDIA GeForce GTX 3080 10GB	32GB 3600mhz

This chapter begins by outlining the libraries needed to implement the machine learning models and carry out the cleaning process discussed in this thesis. Next, the justification for not implementing clinical NER from an experimental standpoint is discussed.

4.1 Libraries used for experiments in this thesis

The list of Python libraries contained in this section outline all of the important libraries that have been used to produce the results in Chapter 6. In addition to the use cases listed here, each library will also contain other machine learning tools that may be relevant for other projects.

- Natural language toolkit (NLTK) (Bird, et al., 2009)
    - Conglomerate library consisting of corpora and resources to provide ease of use access to text processing tools such as (but not limited to) tokenisation, part-of-speech tagging and lemmatisation. In this thesis the NLTK library has been used to retrieve an initial English language stop-word corpus.
- Regular expression (RegEx) (Aho, 1991)
    - Python implementation of regular expressions that rely on a series of characters to create a unique search pattern. These expressions are used to find and replace unwanted information contained within the base letter provided by the general practitioner, removing elements such as punctuation and added whitespace.
- SciPy (Virtanen, et al., 2020)
    - Scientific computing package containing other packages such as NumPy, Matplotlib and pandas. Within this thesis the SciPy library has been used directly for the saving and loading of sparse TF-IDF matrices before and after vocabulary reduction.
- NumPy (Harris, et al., 2020)
    - Numerical computing library which provides the syntax for manipulating arrays, vectors, and matrices. The NumPy library is included specifically in this thesis to convert data into usable arrays and remove specific rows and columns from a sparse matrix. However, NumPy forms the basis for computations carried out by other libraries discussed in this chapter shown below.
- pandas (McKinney, 2010)
    - The pandas library provides an easily analysable and manipulatable set of data structures when working with both numeric and text data. For this thesis, pandas has been used to hold the initial set of

    - documents and their associated tags for pre-processing, and later used to compare the predicted outputs to the expected values.
- scikit-learn (Pedregosa, et al., 2011)
    - Machine learning library built to provide access to tools in data driven analysis. This thesis implements features from the scikit-learn library to classify the text data and print the results either as an accuracy score or in a readable classification report format.
- TensorFlow (Abadi, et al., 2015) and Keras (Chollet, 2015)
    - Open source machine learning platform. This thesis implements both to create the neural network architectures used in 5.2.2 and 5.3.2.
- HuggingFace (Wolf, et al., 2019), Simple Transformers (Rajapaske, 2019)
    - Open source library of pre-trained models and implementation tools for Transformer models used for classification in 5.3.2.2.
- LIME (Ribeiro, et al., 2016)
    - Local Interpretable Model-agnostic Explanations is a library that produces human readable outputs about modelling decisions. The example shown in 5.4.2.1 uses the LimeTextExplainer module is used to create a simplified linear model from the results of the original support vector machine. This new model can then be used to display human readable outputs of the relationships between features in a document and the predicted label.

4.2 Datasets used in this research

As shown in Chapter 3, there is a wealth of data modelling techniques that can be applied to free text data, both inside and outside of the medical field. It also explained the issues that can occur when using medical data with lack of availability and the inclusion of a narrow expert dataset. This chapter outlines the three datasets used to produce the results present in this thesis that avoid the issues listed in Chapter 3. All of the data used was extracted directly from DHCW and are all actual records of patients at varying stages of a clinical pathway. Whilst the dataset outlined in Section 4.2.2 is a curated dataset, the separation of this data was completed before the beginning of this project.

4.2.1 Emergency Department Dataset: Presenting Complaints

The dataset used for classification purposes was the Emergency Department Dataset (EDDS). The EDDS contains information about hospital admissions that come through the Accident and Emergency department. The data being used for classification is the presenting complaint field, a short form string inputted into the system by a receptionist as the patient arrive in Accident and Emergency. It relies on the description of symptoms passed from the patient to the non-medical personnel before a nurse or doctor triages them.

Table 10 Fields contained within the DHCW Emergency Department Dataset

Field Name	Description
Unique Key	Alphanumeric String that is unique to each event
Presenting Complaint	Plain text description of the problem given by the patient on arrival in Accident and Emergency
Treatment Speciality	Name of the Speciality that the patient was assigned to.
Date of Incident	Date and time of incident: dd/mm/yy h:m:s
Consultant Speciality	Speciality of the consultant assigned to patient

Table 10 outlines the fields that were present within the final dataset. The combination of the unique key identifier and the date of incident was used to create a set of unique events. Initial exploration into the data found that the consultant speciality that was listed in the table was unlikely to correlate directly with the outcome treatment speciality. This is due solely to how an Accident and Emergency department is run. The clinicians will have to deal with patients across various specialities rather than having individuals assigned/referred to them. Instead, the focus of classification tasks carried out on this dataset is to extract any relationships between the presenting complaint text field and the treatment speciality.

The EDDS dataset consists of 839,330 events spread across eighty-two medical specialities which causes potential issues with dimensionality. The large amount of vocabulary terms stretched across the eighty-two classes can lead to extremely sparse data with no clear boundaries to separate between classes (Debie & Shafi, 2019).

The separation of these events into categories is shown in Figure 4.1. There is a significant imbalance of representation between the classes in the dataset. This could potentially mean that the accuracy of any model used to classify the data will be lower than if the classes were balanced, especially considering that forty-five out of eighty-two categories have less than one hundred examples. The full table of frequencies for each of the eighty-two medical specialties can be found in Appendix 1.

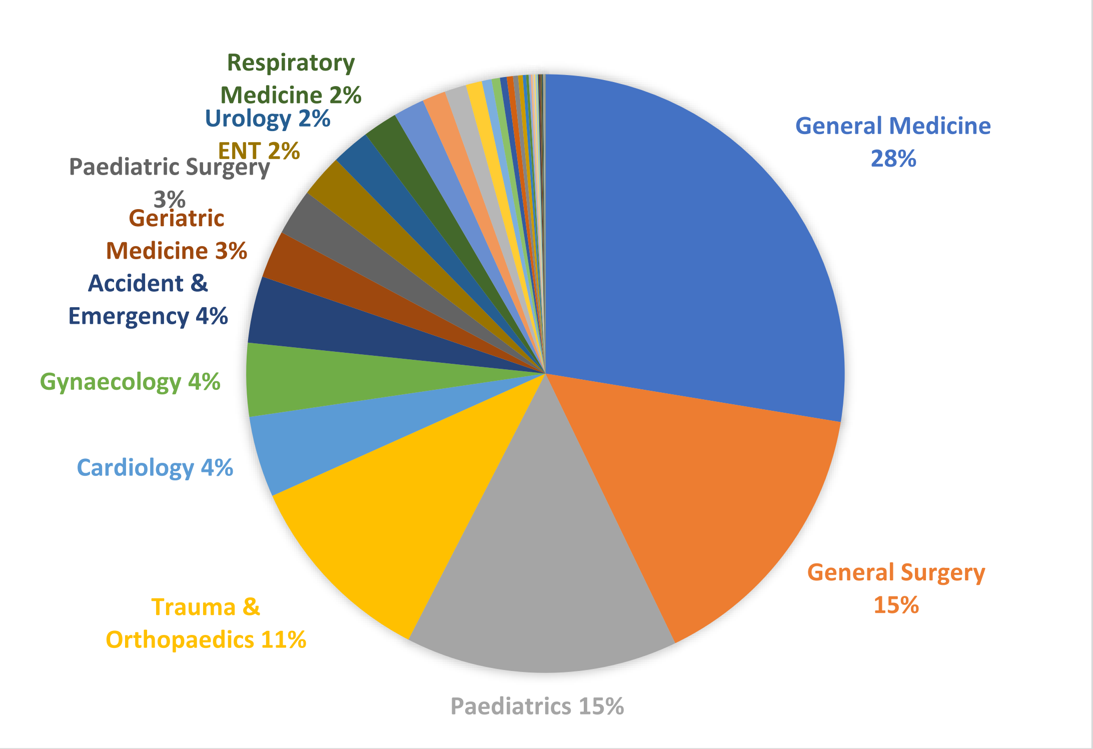


Figure 4.1 Emergency department dataset medical specialty distribution

A subset of this data with the general surgery and general medicine classes removed was also used for classification testing. These were removed through communication with members of staff at DHCW because these two classes may be potentially used as a universal class for placing patient cases if a more specialist class is not obvious.

4.2.2 Urology letters dataset

The urology dataset is an already curated series of general practitioner referral letters to a hospital within the speciality of urology alongside associated labels. The dataset comprises 35,156 unique entries, each consisting of the following fields shown in Table 11.

Table 11 Fields contained within the DHCW Urology dataset

Field	Description
Letter	Plain text letter from the doctor to the hospital consultant
DocumentID	Unique alphanumeric sequence for each document
PrioritisationData	Pre-programmed DHCW database entry which includes clinical coding and patient subcategory diagnosis
GP	Priority given by the general practitioner to the patient being referred
Consultant	Priority is given by the hospital consultant to the patient being referred

The first stage of preparing the dataset for classification was to look carefully at the *PrioritisationData* field and decide how to extract the diagnosis and clinical codes needed to label each document. Each of these fields attributed to a document was different and held varying levels of detail. Two examples of this field are shown in Figure 4.2 to depict the difference in depth of detail contained within the dataset.

2000037|1|Consultant|CONSG|Mixed Clinicians~ 2000037|2|Clinical Condition|UR06|PSA

3000062|2|Clinical Condition|352|PSA

Figure 4.2 Examples of PrioritisationData Field

Each section of information (separated by a vertical bar) is manually inputted into a database by a DHCW employee who received the letter from the relevant hospital health board. The information looked for in this project is the description of the condition/diagnosis at the end of the field and the alphanumeric code preceding it. Individual string cleaning was carried out by splitting them into arrays via the vertical bar marker whilst only keeping the data in the final two entries. This allowed for the data needed for the project to be collected correctly without worrying about the length of the array created due to the varying sizes of each prioritisation string. As seen in Figure 4.2, even when the ending condition is the same, not all the clinical codes present within the database match each other. This may occur as no singular coding standard is shared across the seven Welsh health boards. This also occurs when the code was missing on the initial letter received by DHCW and the resulting input code for the database is the same as the condition (e.g., PSA|PSA).

Table 12 Additional health board codes combined into larger urology labels

New codes	Combined codes
UR01	male luts
UR04	348, 349, loin pain stone
UR05	haematuria
UR06	psa
UR07	uti
UR09	346, 744, 745, 746, 347, erectile dysfunction

A rule-based system was implemented to correct this labelling issue, converting the incorrect labels into usable codes for nine distinct categories, UR01-UR09, as shown in Table 12. These rules were created through the use of the NHS data dictionary (NHS Wales, 2020) and communication with DHCW clinical coding staff. While this rule-based system worked for approximately ninety-five percent of cases, some manual inspection and correction was needed in situations where the condition was more detailed than expected. An example of this is where whoever coded the information included “Haematuria – visible” instead of just haematuria. Within the dataset there were several cases with speciality labels belonging to a small group that have since been removed. There was a combination of examples labelled as “Other/Unknown”, “SysOther”, or with test codes. These have also been removed, reducing the dataset size to 25,451. Table 13 shows how the letters are split into nine categories after extracting the labels from the prioritisation data field.

Table 13 Extracted codes and conditions from the PrioritisationData field within the dataset after cleaning

Code	Condition	Supporting Documents
UR01	Male lower urinary tract symptoms	3869
UR02	Female lower urinary tract symptoms	650
UR03	Penoscrotal	1763
UR04	Loin pain and stone	1631
UR05	Haematuria	8386
UR06	Prostate specific antigen (PSA)	5549
UR07	Urinary tract infection	2448
UR08	Cancer	366
UR09	Andrology	789

With the documents categorised correctly, the next step was to clean the letters themselves. Cleaning the documents aims to alleviate issues that are created from human error in order to create better inputs for the Natural Language Processing pipeline. The dataset was loaded from a .csv file into a pandas dataframe using only the columns for the letter and code. Each document was then cleaned as follows:

1. tabs removed
2. remove punctuation and numbers, leaving only alpha characters
3. remove multiple spaces
4. remove leading and trailing spaces
5. convert each word within a document to lower case
6. replace the existing document with the cleaned version

The author decided to keep all extracted terms present in full due to the nature of certain words having different meanings within a medical context. No use of stemming or lemmatising has been carried out on the dataset. The last step taken to prepare the data for vectorisation was to include a list of stop words to remove potential redundancies from the dataset. The system utilised the English stop word corpus from NLTK (Bird, et al., 2009). The author then extended the standard list to better meet the project's requirements (shown in Table 14).

Table 14 Stop words added to the NLTK library for the Urology Dataset

dear, colleague, sincerely, grateful, thank, thanks, seen, review, history, today, requested, ago, dr, please, would, could, also, examination, old, year, patient, help, advice, management, opinion, see, last, months, however, given, years, may, since, symptoms, sent, well, due, showed, appreciate, regarding, presented, kind, regards

The words in the extended list mainly consisted of variations in salutations and valedictions that would be expected in a full letter. The other stop words included were ones like advice and opinion, words that convey the conversational aspect of the letter between the two medical practitioners rather than any marker towards the condition of the patient.

4.2.3 General Practitioner Referral Letters

The final dataset discussed in this chapter is another set of referral letters between a general practitioner and a hospital consultant. These referral letters were selected at random from a DHCW database with no prior of knowledge of what speciality or priority they had been assigned. Initially the size of this dataset was 121,146 documents that was reduced to 111,128 due to the cleaning process described in Section 4.2.3.1. Table 15 contains the list of fields within this dataset. The primary fields concerning this thesis are the ExtractedText field which contained the unedited letters, the speciality, and the priority fields.

Table 15 Fields contained within the DHCW multiple specialities dataset

Field	Description
ExtractedText	Plain text letter from the doctor to the hospital consultant
DocumentDateTime	Date/time the document was put into the system
EventDateTime	Date/time of the GP appointment
VersionNumber	How many times the letter has been passed between GP and consultant
SubjectSexCode	Sex of the patient (M or F)
Speciality	Name of the Speciality being referred to
GPPriority	Priority given by the general practitioner to the patient being referred
ConsultantPriority	Priority given by the hospital consultant to the patient being referred
DateReferred	Date of the GP appointment
PatientReferralAge	Age of the patient
LocalHealthBoardResidenceCode	Associated code with the Welsh health board
LocalHealthBoardResidenceName	Associated name of the Welsh health board

Figure 4.3 shows the associated version numbers for each letter in the dataset. The version number denotes a round of correspondence between general practitioner and consultant. Increasing the version number by one means that something has occurred with the case. This may be that the case is forwarded on to a different medical specialist, new test data has been added, or an acknowledgement/rejection of a change of priority has been made for the case. The data shows that through random sampling of a wider database, at least fifty percent of all cases undergo changes. Each change made to a referral increases the timeframe before the patient receives

their appointment letter and the amount of work both clinicians are required to spend on a case. The ability to classify documents correctly could ensure that a first stage version change because of medical speciality or prioritisation would no longer be needed.

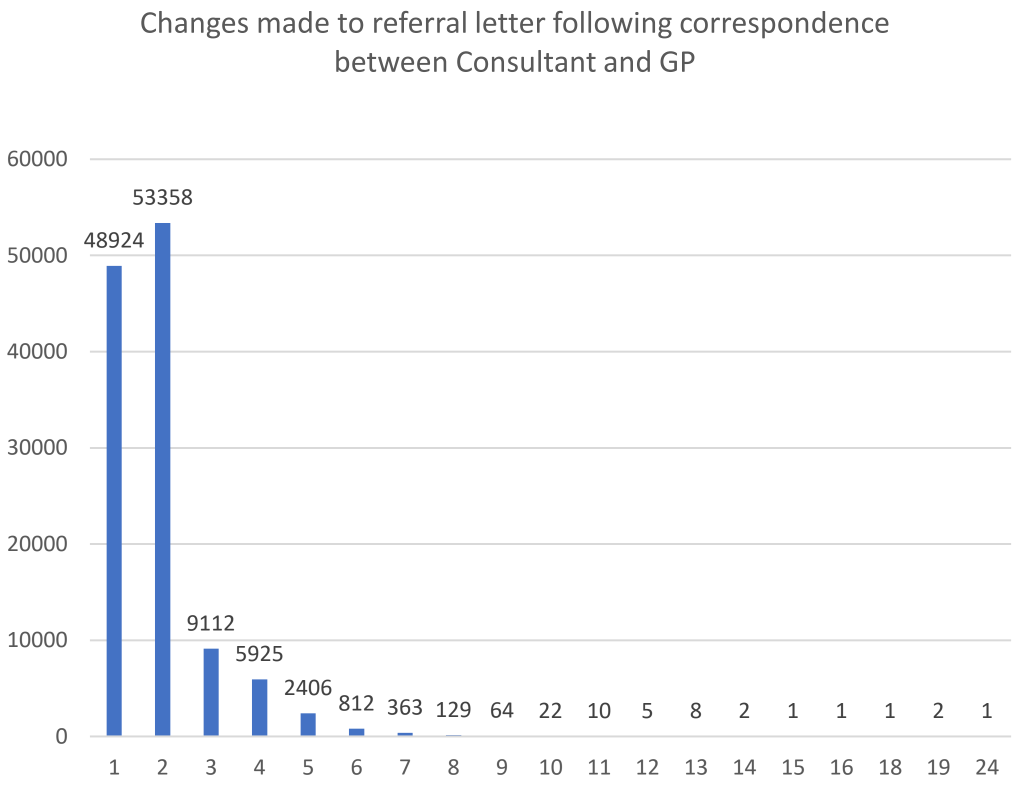


Figure 4.3 Version numbers showing the number of changes that have been made to a referral letter by a GP or consultant in the dataset

The following outlines the process taken to extract a usable set of data from the *ExtractedText* field and the two classification outcomes pursued with this dataset. The first classification relating to the medical speciality assigned to a letter and the second related to the assigned patient priority.

4.2.3.1 Dataset cleaning

The free text analysed in this thesis exists as a part of a larger document. The content of this document lists a variety of clinician and patient-related data, including personal information that needed to be excluded before the data could be processed. This information is contained within Figure 4.4.

A manual inspection of the letters found useful headings that could be used as indicators for the start and end of the main body of the referral letter that was important to the project. After reducing the letter to the information between these headings, variations on salutations were used to find the starting point of the letters, and the final token before a valediction was used as the ending point.

General Practitioners priority

Patient Details – Name/sex/Address

GP Details – Code, Practice name/code/Address

Consultant Details – Speciality/Hospital site

Reason for Referral

Care Type Requested

Expected outcome

Problems/Diagnoses – Description and dates of previous diagnoses

Operations/Procedures – Same as above, also includes new patient screening

Referral Speciality

Date of Referral

Hospital

Urgency

Presenting complaint – Free text by GP explaining the situation

Lifestyle information (not always filled in) – History of alcohol/exercise/smoking

Medical History

- Recent Medication
- Measurements (height/weight/Blood pressure)

Any additional relevant information – i.e., Veteran status

Figure 4.4 Outline of a referral letter

The same process as the one described in Section 4.2.2 for the urology dataset was followed for the cleaning of the referral letters themselves. Within a small number of letters, it was found that there was a letter within another letter as it was a repeated referral. The maximum number of words within a letter was set to 251 to counteract these abnormalities in the data. Similarly, the minimum number of words required to be considered as an actual letter was set to 10 to remove instances where the letter consisted of a variation of the phrase "please find the attached referral". A limited, anonymised subset of cleaned letters can be found in the University of South Wales Computer Science and Artificial Intelligence Paradigms (CSAIP) research repository[1]. The stop words list from the previous dataset was again implemented to help reduce the impact of conversational noise within the letters themselves. However, it was found that the default stop word list from the NLTK library (Bird, et al., 2009) contained the words "no" and "not". These two words were removed from the stop word list as the difference between "signs of symptom" and "no signs of symptom" would impact the speciality pathway a patient may take.

4.2.3.2 Medical specialisms

The information for specialties was already separated into a separate field instead of being contained within a single string of information like in the urology dataset. However, due to the increased size of the dataset, there were more variations in the speciality field. Without any data pre-processing, the dataset consisted of 108 different specialities and documents of sizes varying between 1 and 14760.

Through communication with members of DHCW and the use of the NHS Wales data dictionary (NHS Wales, 2020), certain specialties have been grouped together. The separations occurred in the original dataset due to health board differences or a specific condition (neurology department compared to neurology focused on epilepsy). The changes made are explained below in Table 16. The name used for the new class already existed within the dataset. Examples without a combined speciality only existed as that version. All classes are represented as presented in the dataset; spelling issues included.

[1] https://intelligence.research.southwales.ac.uk/documents/edit/3573/Letters.zip
(Password: ESSSbMRLuaSVM)

Table 16 Grouping of specialties into more concise classes

Specialities	Combined Specialties	Supporting Documents
Cardiology	Cardiology (card)	3606
Care of the elderly	Care of the elderly usc identi	275
Clinical Immunology		444
Clinical neurophysiology		225
Community orthopaedic		2047
Dermatology	Dermatology (derm) Dermatology (usc) Dermatology laser Dermatology usc identifer	14760
Dietetics	Dietetics (dthe)	1682
Endocrinology	Medical endocrinology (mendoc) Endocrinology usc identifer Endocrinology usc identifier	989
Ent	Ent (usc) Ent audiological medicine (entam) Ent usc identifier Ear nose and throat (ent)	12529
Gastroenterology	Gastroenterology (gastro) Gastroenterology (usc) Gastroenterology usc identifer	5210
General medicine	General medicine (genmed) General medicine usc identifer General medicine nurses (gmedn)	1098
General surgery	General surgery (surg) General surgery usc identifier General surgery breast clinic (surb/c) General surgery breast service Breast (usc) Breast Gs breast usc	14312
Geriatric medicine	Geriatric medicine pathy day hosp (gerijp) Geriatric medicine (geri)	406

Gynaecology	Gynaecology (gynae) Gynaecology (usc) Gynaecology usc identifer	10182
Haematology (clinical)	Haematology (clinical) usc ide Haematology-clinical (haem)	720
Nephrology	Nephrology (neph)	436
Neurology	Neurology (neur) Neurology epilepsy (epilep) Other neurology Other neurology usc identifer Other neurology usc identifier	2172
Oral/maxilla facial surgery	OMF usc identifer Oral/maxilla-facial surgery (oral) Omf usc identifier	1039
Ophthalmology	Ophthalmology usc identifer	719
Orthopaedic	Orth foot & ankle (t/ofa) Orthopaedic hand (t/hand) Orthopaedic hip (t/ohip) Orthopaedic knee (t/knee) Orthopaedic paediatrics (t/paed) Orthopaedic shoulder (t/osh) Otrhopaedic spinal (t/osp) Orthopaedic spines Orthopaedic spines usc ident Trauma & orthopaedic Trauma & orthopaedic usc ident Trauma & orthopaedics (t/o)	11408
Paediatrics	Paediatric endocrine (pendo) Paediatric gastroenterology (pgast) Paediatric respiratory (presp) Paediatric cardiology (pcardl) Childrens ent (entpae) Paediatrics usc identifier Paediatric surgery (paedsu)	3646
Pain management	Chronic pain management(chronp)	807
Physiotherapy adult		7072

Rapid diagnostic centre	Rapid diagnostic centre usc	152
Rehabilitation	Rehab day hospital (rehadh) Rehabilitation (rehab)	656
Rheumatology	Rheumatology (rheum)	2141
Thoracic Medicine	Thoracic medicine (throme) Thoracic medicine usc identif Respiratory (usc)	2912
Urology	Urology (usc) Urology (urol) Urology usc identifer	8706
Vascular surgery	Vascular	777

These combinations resulted in twenty-nine usable specialties for the purpose of classification. The additional specialties existing within the dataset are shown in Table 17. These specialities were unable to be combined with other specialties and were removed due to the lack of supporting documents.

Table 17 Removed specialties due to limited supporting documents.

Speciality	Supporting Documents
Adult speech and language (salt)	59
Occupational therapy	36
Clinical pharmacology	73
Stroke medicine	37
Oral medicine	28
Tier 3 weight management	19
Gender services	18
Plastic surgery	2

4.2.3.3 Patient Prioritisation

The secondary classification task surrounds the priority assigned to a patient's case as they get transferred to a hospital. Three priorities exist within the dataset: Routine, Urgent, and USC (urgent suspected cancer). These priorities are consistent across the health boards in Wales and require no label cleaning like the specialisms in 4.2.3.2.

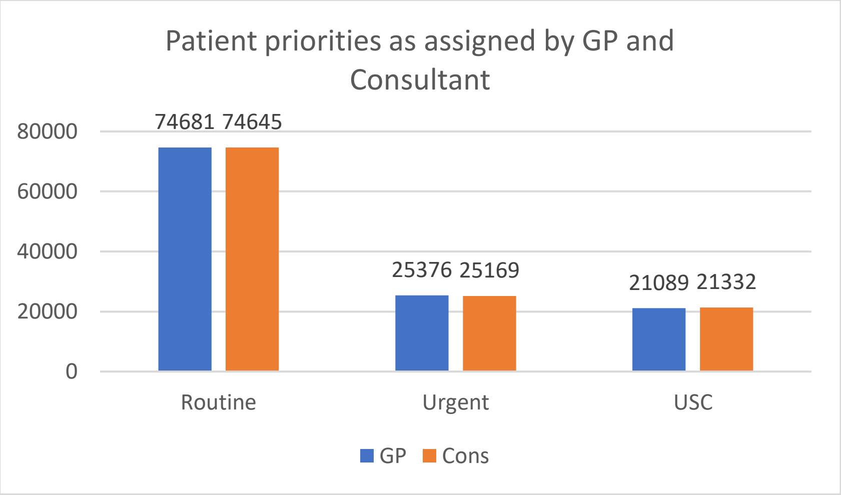


Figure 4.5 Assignment of patient priority across the whole dataset

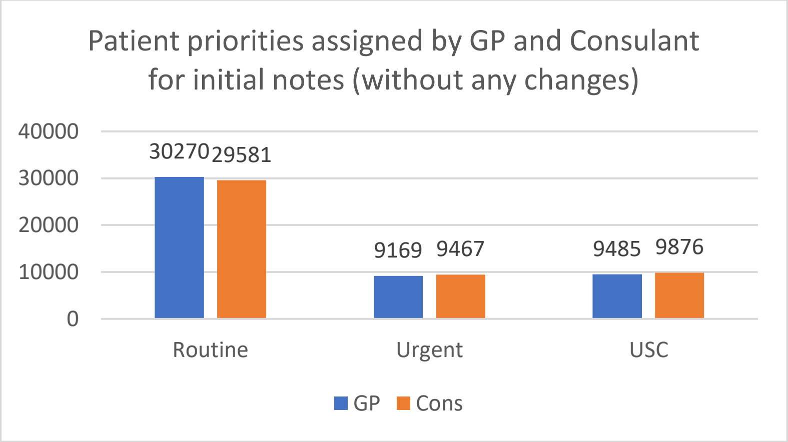


Figure 4.6 Assignment of patient priority specific to initial referral

Figure 4.5 and Figure 4.6 show the number of patient cases assigned to a specific priority. In Figure 4.5, there are only minor differences between what a general practitioner assigns and what a consultant will assign when encompassing all of the dataset. The largest differential is in the USC class with a 1.1% higher likelihood that a consultant would mark a case as USC. Figure 4.6 however begins to outline the difference in opinion a general practitioner will have to the consultant when first referring a patient to hospital. The differentials for the same classes are multitudes higher than they were in the first view (Figure 4.5).

However, the bar graph representation of this data only reveals a portion of information regarding the issue with prioritisation. Using the graphs alone does not show the number of prioritisation changes that occur in the initial referral process. There are two confusion matrices shown in Figure 4.7 that describe these differences.

Figure 4.7 (a) shows a large amount of movement between prioritisations when a consultant reads the initial letter. These changes occur as both downgrades and upgrades. Given the referral letters analysed in this thesis a general practitioner's assigned priority will only match the consultant's priority for an initial referral 83.67 percent of the time. The biggest movement in priority is seen in the urgent class. Only 73.65 percent of cases listed as urgent by the general practitioner are kept as urgent. The other 26.35 percent are either upgraded to USC or downgraded to routine.

Figure 4.8 (b) shows a stark difference for referrals that have undergone a second iteration between clinicians. The assigned priorities from both clinicians are more likely to match than the original referrals (98.9 percent). While the urgent class is still the most volatile out of the three classes, the disagreement between clinicians is significantly lower. What this suggests is that the first version update of a referral letter is all about the consultant adjusting the priority assigned to a case.

The goal of this classification task is then to target this initial disparity between the general practitioner and consultant priority. If a classifier can map the relationship between the words in an incoming referral letter and what priority a consultant will assign, a sizeable portion of correspondence could be removed. Again, the choice of output is important. A choice could be made for auto-prioritisation for cases above a certain confidence threshold. However, taking the same direction as the medical specialism classification, providing a support tool that the general practitioner can see how a letter has been prioritised (highlighting key words and phrases) is also considered.

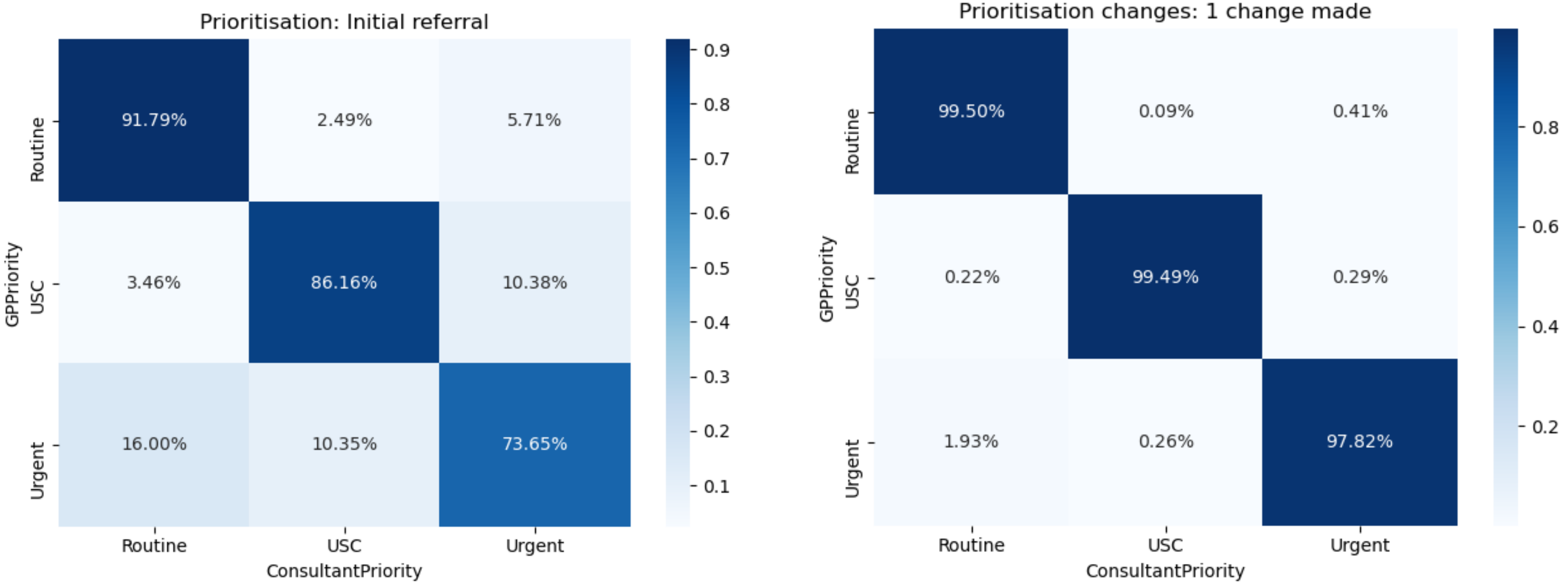


Figure 4.7 Patient Prioritisation matching between GP and Consultant when either (a) no changes or (b) one change is made

4.3 Machine Learning Algorithms

This section summarizes the different machine learning algorithms that have been used to produce the results shown in Chapter 5 of this thesis. It has been broken down into three sections: unsupervised learning with clustering algorithms, traditional machine learning algorithms and neural networks.

4.3.1 Unsupervised Learning: Clustering

Clustering is an unsupervised approach to data grouping, separating data into clusters based on similarity. Whilst the classification methods described in this section require a set of labelled training examples to assign roles to data, an unsupervised approach relies solely on the implemented algorithm to separate data into groups of information. Using an unsupervised method like clustering can provide a good, data-driven opportunity for discovering initial insights into a dataset even when the dataset is labelled. Within the medical subdomain, work done by Patterson & Hurdle (2011) and Dong-Harris, et al. (2013) speak to using k-means clustering for medical specialities and how it can benefit datasets that have different labels due to the information coming from a variety of sources.

The clustering algorithm used in this research is the k-means++ clustering algorithm (David & Sergei, 2007). It is an adaptation to the k-means algorithm that employs a directed initial centroid placement strategy to ensure centroids (points chosen to be centres of clusters) are evenly distributed across the dataset. Once the centroids have been placed, the k-means++ algorithm uses a squared error function (Lloyd, 1982) as its objective function (7) to determine which cluster each document belongs to.

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \tag{7}$$

Where:

- J is the objective function
- k is the total number of clusters
- n is the number of documents (cases)
- $x_i^{(j)}$ is the document current being placed
- c_j is the centroid for cluster j

4.3.2 Traditional Machine Learning algorithms

The term traditional machine learning algorithms encompasses the different models available for tasks such as document classification that do not employ a neural network-based approach to training. Some of the prevalent models in this category include support vector machines, logistic regression, k-nearest neighbours, and decision trees. These models have shown to have comparative accuracies to the neural network and transformer based architectures on similar tasks (Behera, et al., 2019), including tasks within the clinical domain such as autism detection (Lee, et al., 2019) and classifying medical specialities (Weng, et al., 2017).

This research compares seven different traditional machine learning algorithms. The hyperparameters for each of the algorithms have been determined through experimental means, with some techniques (random forest classifier) requiring a significant increase in time and effort to achieve similar results to the other methodologies. Table 18 shows the traditional machine learning techniques and their associated hyperparameters used in the comparison experiments. Each of the models can be accessed through the sci-kit learn (Pedregosa, et al., 2011) Python library.

Table 18 Models and parameters used for K-fold cross validation on GP letters

Model	Parameters
Linear Support Vector Machine	Penalty : l2 Loss: squared_hinge C : 1.0
Support Vector Machine (RBF Kernel)	C : 1.5 Gamma: 0.5
Logistic Regression	Penalty : l2 C : 1.0 Solver : lbfgs (Malouf, 2002)
Stochastic Gradient Descent (SVM equivalent loss penalty)	Penalty : l2 Loss : hinge Alpha : 0.0001

Random Forest Classifier	Estimators : 100 Max_depth : 80 Max_features : sqrt(features) Split Criterion : Gini Impurity
Bernoulli Naïve-Bayes	Alpha : 1.0 Class Priority : None
Multinomial Naïve-Bayes	Alpha : 1.0 Class Priority : None

4.3.3 Artificial Neural Networks

Artificial neural networks (ANNs) are computing systems used in machine learning built around replicating the transference of knowledge via signalling that occurs via neurons in an animal brain. A neural network trains through experience and iterative learning, updating a series of small weights and biases until the desired result is achieved. This desired output of a neural network model will depend on the problem tasked, as neural networks are used for a variety of tasks, including classification and pattern recognition.

Over the years, several types of neural network architectures have been created to address the nuances of specific machine learning tasks like using a convolutional neural network for image recognition. However, these architectures are still able to produce outputs that may match or improve on the accuracy of traditional machine learning methods in other tasks like document classification (Rabhi, et al., 2019). This performance was echoed for the i2b2 (Uzuner, et al., 2008) challenges discussed in Section 2.4 where deep learning architectures were able to outperform the models from the original challenge winners (Lee, et al., 2020; Zhang, et al., 2021). However, the same architectures have shown to have issues when classifying to a larger number of classes (Tang, 2015) than a simpler binary classification.

This research employs linear, convolutional, recurrent, and transformer-based neural network architectures to produce the results shown in Chapter 5. The choice of optimiser for training these networks was adaptive moment estimation or Adam (Kingma & Ba, 2014). Adam is commonly found as the default optimiser choice in newer libraries of neural networks, especially those using newer architectures such as transformers (Wolf, et al., 2019). For the results shown using transformer-based architectures, RoBERTa-base (Liu, et al., 2019) and BioBert (Lee, et al., 2020) were used to contrast models trained on general data versus biomedical documents.

4.4 Experimental reasons for excluding clinical named entity recognition using existing pre-trained libraries

The results shown in Chapter 5 have been produced without additional clinical annotations being overlayed onto the data. This decision was reached based upon a combination of previous literature and a test of external clinical NER implementations on unseen medical letters. As noted in Section 2.4, the incorporation of one of these pre-trained software packages is commonplace within clinical literature. However, it was shown through articles by Weng et al. (2017) and Gehrmann et. al. (2018) that the method used to transform the data into input vectors or word embeddings has a larger impact on the ability of the system to accurately classify the data than the inclusion of clinical annotations.

Three open-source packages are available for clinical named entity recognition that could have be included in a project and will be explained further in this section. These packages are: Apache cTakes (Savova, et al., 2010), MetaMap (Aronson, 2001) and QuickUMLS (Soldaini & Goharian, 2016).

To determine the effectiveness of these available software packages, two randomly selected doctors’ notes were taken from a DHCW database and analysed. As shown in Figure 4.8, the quality of grammar and overall sentence structure will depend on the author of the letter and may also vary on a case-by-case basis from the same doctor. The input for each annotator was a single string for the entire letter, no external tokenisation was done as each annotator was a pipeline with this feature built in. The rest of this section will outline the results of applying these annotators to this data.

Case 1 – *I reviewed <patient> who is struggling to hear conversation and also with watching tv. She has bilateral non intrusive tinnitus which she can mainly hear at night. Clinically both her ears appear normal and pta showed mild deterioration of her hearing mainly on the right when compared to her audiogram in 2017. However, her hearing thresholds remained stable compared to her audiogram in 2016. I have discussed the role of amplification and she has opted for bilateral hearing aids. Though there is some asymmetry in her hearing, I feel the best way forward is to monitor this and I have arranged to review her again in one year. Yours sincerely.*

Case 2 – *post-op removal of plate from 1st tmt fusion 19-jun-2018 this lady is now 2 weeks following her surgery. Her wound has healed beautifully with no evidence of infection and the sutures have been removed today. She did have a fall on a bus recently but thankfully didn't come to any harm. I have advised her to get into normal footwear and gradually get off the crutch. We have given her another 4 weeks of enoxaparin due to her previous dvt. We will see her back in the clinic in 4 weeks time. With kind regards*

Figure 4.8 Case study for comparing NER packages

4.4.1 Apache cTakes

Apache cTakes annotates eighteen different examples from the first case, including the terms *tinnitus* and *audiogram*. However, there are number of issues with the output that dispute the effectiveness of using it as a blanket annotator across the large datasets in this thesis. On multiple occasions individual words were picked out as terms instead of being combined into a longer medical concept. In case one the medical device *hearing aids* was classified as two different annotations: *hearing* and *aids*; the second having no relation to anything present in the case study.

The inclusion of medical abbreviations in the texts also produces issues. In case one the medical abbreviation *pta* is present and given the context of this letter, pta stands for pure tone audiometry. The annotator labelled the abbreviation as plasma thromboplastin antecedent instead, a coagulation factor, which would direct the output of a classifier towards clinical haematology and away from the ENT speciality. No alternatives were offered as to the meaning of the abbreviation.

The second case shows an issue with the wide nomenclature of medications. The word *today* was mislabelled twice as the antibiotic ingredient cephapirin sodium used for veterinary means which has the preferred name ToDAY (Whetzel, et al., 2011). This followed the word *bus* being extended to Busulfan instead of attributed to the vehicle the patient fell down in.

4.4.2 MetaMap

MetaMap produced a better overall annotation of phrasal features in the case studies than the cTakes annotator. Phrases such as *bilateral non intrusive tinnitus* in case one and *post-op removal* in case two were annotated as one detailed feature. The phrase *bilateral hearing aids* was also annotated together unlike previously seen in Section 4.4.1. The MetaMap output included a type assigned to each annotation such as finding, location and laboratory procedure.

Uncommon medical abbreviations proved to also be an issue with MetaMap. The *pta* abbreviation was expanded and annotated as prothrombin activity measurement, which is another medical concept that does not relate to the same speciality as the original letter. MetaMap annotated non-medical terms successfully. Bus was annotated as a location for the patient’s fall and temporal concepts within the letter (today, now, recently) were also annotated.

4.4.3 QuickUMLS

The QuickUMLS annotator was the fastest but least robust annotator. It produced the least number of annotations on the case studies overall. However, it does offer a list of alternatives and their associated confidence levels. While some phrasal information has been retained, the performance lacks behind MetaMap.

The annotator’s approach is an approximate dictionary matching algorithm, meaning the annotator can only provide a similarity score to those topics in the dictionary. Any general concepts, like *bus*, present within the case study had no annotation attached. Similarly, none of the abbreviations were expanded and annotated.

4.4.4 Observations

The three annotators exemplify the issue with Natural Language Processing on medical data. Attempting to map terms in a letter to broad medical dictionaries results in irrelevant annotations that obscure outcomes. While this approach has shown to have minor improvements to the overall accuracy of a classification model, it would reduce trust in a system where the results of individual terms are outputted to the screen. Requiring a clinician to perform manual verification of every term annotated would provide no benefit in usefulness or work reduction.

4.5 Chapter conclusion

This chapter steps taken to best represent the relationships present within the data and therefore produce machine learning models that have acceptable accuracy scores. The steps detailed in Section 4.2 provide a unique perspective towards cleaning and preparing a medical dataset that exists without a reliable structure. It further explains the pitfalls of employing existing libraries thoughtlessly with both generic stop word lists and medical ontologies. Section 4.3 outlines the different types of machine learning models being used in this research for both supervised and unsupervised learning. This chapter also provided an experimental reason for not depending on open-source clinical annotators such as cTakes. It shows that there are errors that can occur if the resulting outputs are not validated. The following chapter describes the experimental results derived from the application of algorithms to the data described here.

Chapter 5 NLP Implementations and Results

This chapter outlines the results produced by clustering and classification methods on the three different datasets described in Chapter 4. The approaches and machine learning tools used on the datasets in each section influence the decisions made with subsequent datasets. These choices culminate in the system detailed in Section 5.4 which draws on the idea of presenting multiple options to a clinician as a supporting tool rather than a direct classification tool that overrides any medical professional's input.

5.1 Results using the presenting complaints dataset

The results produced using the presenting complaints dataset were used in communications with DHCW as an indication that machine learning algorithms could be beneficial with patient data but required a more sophisticated dataset to train on to achieve better results. As explained in 4.2.1, the presenting complaints dataset was made up of short, unstructured lists of symptoms that a person was describing to the receptionist at an accident an emergency department whereas the later datasets were full bodied doctors' letters.

The research was carried out using supervised classification methods alone (outlined in Section 3.4). A 5-fold cross validation strategy has been used to determine an estimated level of accuracy between the models on the presenting complaints data. Table 19 outlines the accuracies found using the whole dataset and the same dataset with the general medicine and general surgery categories as outlined in Section 4.2.1.

Table 19 Classification accuracies using 5-fold cross validation

Model name	All categories accuracy	Subset accuracy
Logistic Regression	.608371	.584923
LinearSVC	.604581	.582112
MultinomialNB	.59436	.567942
Random Forest	.32544	.232453

The results in Table 19 show that the classifiers used with this data were able to discover some of the relationships present between the input vectors and the labelled medical specialities. Three out of four classifiers produced results that exhibited signs of generalised learning, while the fourth struggled to accurately portray the data. The accuracy reported by the random forest classifier is around the level of accuracy a person could achieve by assigning every presenting complaint to the general medicine class (.282322). The subset accuracy shows that removing general medicine and general surgery because it is potentially used as a blanket label by accident and emergency staff is unfounded. If this hypothesis was true, the expected accuracy values would increase as the largest classes of data are no longer obscuring the boundaries between other, smaller classes. Instead, there is a general decrease in the model accuracy as the influence of smaller classes has a proportional relation with the number of documents.

The accuracy metric has a significant shortcoming when portraying a dataset with unbalanced classes. There is no penalisation of misclassified documents. Table 20 demonstrates the ability of the models to classify with different performance metrics. Every model's performance shows a greater recall score than a precision score, indicating that each model is presenting more false positives than false negatives. Again, this performance indicator can be attributed to the unbalanced classes in the dataset. If the majority of documents are assigned to the largest classes in the dataset, the number of false positives will be higher than the false negatives.

Table 20 Performance metrics for models on the presenting complaints data

Dataset	Model	Precision	Recall	F1-Score
All Specialties	Logistic Regression	0.59	0.61	0.56
	LinearSVC	0.57	0.61	0.56
	MultinomialNB	0.57	0.60	0.54
	Random Forest	0.26	0.33	0.16
General Medicine and General Surgery Removed	Logistic Regression	0.56	0.58	0.55
	LinearSVC	0.55	0.58	0.55
	MultinomialNB	0.54	0.57	0.52
	Random Forest	0.27	0.23	0.10

The classification models did not achieve a suitable accuracy for use as a system within a clinical environment. However, the models demonstrated that there is a clear relation that can be computationally modelled between signs and symptoms and a medical speciality. It is reasonable to suggest that the reason behind poor accuracy values in this scenario is because of the dataset. The information present within a presenting complaint is an account of the issues from the patient themselves. Secondly, the documents are written in a shorthand format, which lends to a significant use of abbreviations. These medical abbreviations can easily be misconstrued as one another, as previously shown with named entity recognition software in Section 4.3. Finally, the complexity of the problem itself has a significant impact. The problem attempts to relate 839,330 documents to eighty-two different classes of medical speciality, including a significant class imbalance.

A potential alternative use for this dataset might be to target significant, high-priority illnesses such as strokes or heart attacks based on patient symptoms. However, this would require creating a far more specialist model than intended for this project.

5.2 Results gathered from the urology dataset

The urology dataset presents a unique opportunity to try and classify a series of referral letters that will have a significant level of overlap due to the narrow scope of the problem. This section first outlines the techniques for initial data exploration through clustering followed by the classification of the documents into nine distinct urology sub-specialities UR01-UR09.

5.2.1 Clustering results

Clustering experiments were performed on this dataset to evaluate the separability of documents as an initial stage before any classification was performed. The first issue encountered with separation was the incorrect labelling of documents due to health board differences as stated in Chapter 4. The initial clustering analysis provided the insight needed to understand the problem and produce the rule-based system for combining categories. Originally there was eighteen sub-specialities of urology coded into the system. The distortion score heuristic has been combined with a k-means clustering algorithm to determine the number of clusters that best represent the data. Figure 5.1 shows that the best compromise between error value and computational time requirements as twelve clusters, a reduction of six.

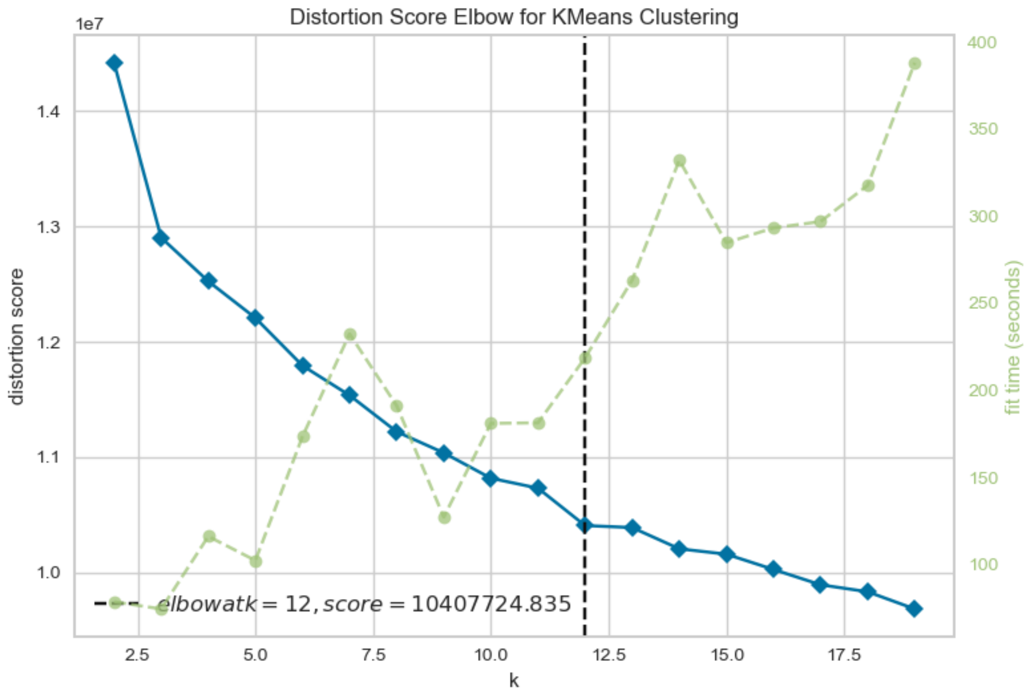


Figure 5.1 Distortion score for the urology dataset with all eighteen classes

While the SSE (Sum of Squared Errors) value is high at 10,407,724 it is related solely to this dataset and cannot be compared to other datasets in published literature. However, it can be compared to variations of the same dataset. The same heuristic process was carried out after the reduction in classes from eighteen to nine specialities (UR01-UR09).

Shown in Figure 5.2, combining several of the categories and removing unwanted training/empty values provides a far better distortion score than originally found. This seventeen percent reduction in error down to a score of 8,661,501 provides a far better representation of the data as the expected best number of clusters is now only one away from the actual number of classes. While taking into consideration the scores provided by the heuristic, earlier work in related literature has shown that the most common method for determining the ability of an algorithm to cluster a dataset is to set the number of clusters (K) to the number of existing classes (K^I) when known (Bradley & Fayyad, 1998; He, et al., 2004; Su & Dy, 2007; Cao, et al., 2009).

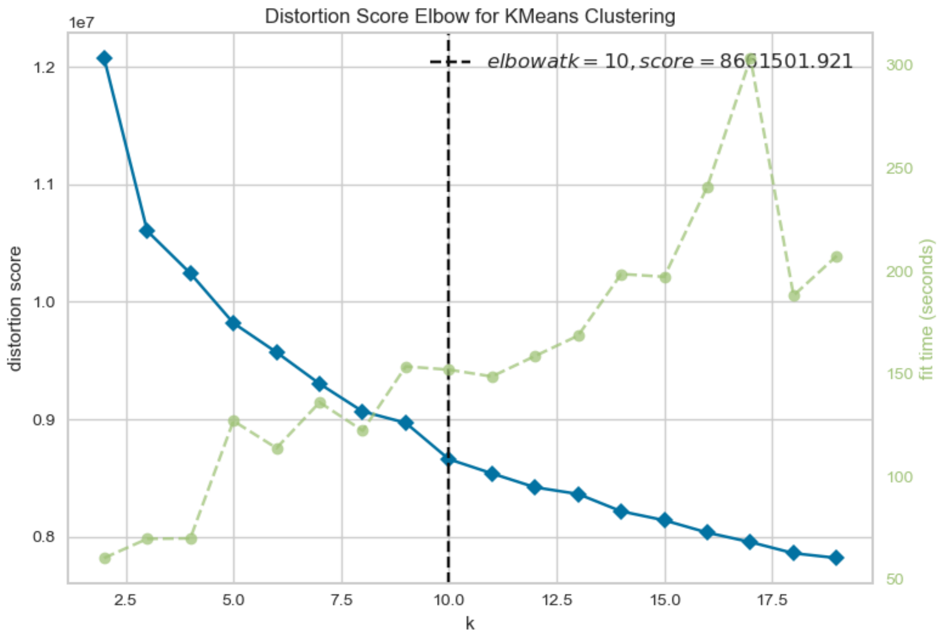


Figure 5.2 Distortion score for the urology dataset with the combined nine classes

Correcting the number of classes within a dataset to the nine subspecialities allows for additional clustering experiments to be carried out. These experiments focus on validating the input of the data into the clustering algorithm: the vectorisation method used and the ability to reduce the feature size.

5.2.1.1 Effect of different vectorisation methods

The vectorisation process of turning free text documents into usable inputs has a significant impact on the quality of clustering and classification. As previously discussed in Chapter 2, there are two main variations on this transformation: Frequency-based vectors and word embeddings.

Table 21 Clustering against nine classes with bag of words vectorisation

Cluster	UR01	UR02	UR03	UR04	UR05	UR06	UR07	UR08	UR09
0	1.3%	0.6%	0.2%	0.4%	0.7%	1.2%	0.3%	2.0%	0.7%
1	0.0%	0.0%	0.0%	0.0%	0.1%	0.1%	0.0%	0.5%	0.0%
2	0.8%	1.2%	0.1%	0.3%	1.5%	0.4%	2.4%	0.7%	0.1%
3	0.0%	0.1%	**99.7%**	0.1%	0.0%	0.1%	0.1%	0.0%	0.0%
4	**97.6%**	**98.1%**	0.0%	**99.1%**	**97.5%**	**97.9%**	**97.1%**	**96.8%**	**99.1%**
5	0.1%	0.0%	0.1%	0.1%	0.0%	0.1%	0.0%	0.0%	0.0%
6	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
7	0.1%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%
8	0.0%	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%

The effects of the different transformation techniques are shown in tables Table 21, Table 22 and Table 23. The k-means algorithm was instantiated with a specific seed (47) to ensure the starting places for the clusters were the same across all experiments. The k-means clustering combined with bag of words vectorisation shown in Table 21 struggles to separate any specific class apart from UR03.

Table 22 Clustering against nine classes with TF-IDF vectors

Cluster	UR01	UR02	UR03	UR04	UR05	UR06	UR07	UR08	UR09
0	2.1%	0.0%	**30.7%**	0.1%	0.3%	1.2%	0.2%	1.7%	**65.0%**
1	2.6%	4.1%	**26.4%**	**73.9%**	4.2%	0.6%	6.5%	**27.0%**	0.7%
2	**55.9%**	2.3%	1.4%	0.4%	3.6%	**84.1%**	1.7%	**35.4%**	2.4%
3	2.3%	1.9%	0.3%	0.9%	2.8%	1.7%	3.0%	2.0%	0.7%
4	2.7%	0.0%	0.6%	2.6%	3.6%	3.8%	2.7%	0.7%	1.6%
5	0.8%	1.6%	0.6%	0.8%	0.8%	1.0%	0.8%	1.5%	0.9%
6	0.0%	0.0%	**23.1%**	0.0%	0.0%	0.0%	0.0%	0.0%	14.3%
7	**33.0%**	**88.6%**	16.8%	16.7%	66.1%	7.2%	**81.6%**	**31.0%**	14.4%
8	0.6%	1.5%	0.1%	4.6%	18.6%	0.4%	3.4%	0.7%	0.0%

The TF-IDF vectors provided a better separation between the sub-specialities in the urology dataset. There are significant overlaps between some of the classes in individual clusters, such as UR03 and UR09 sharing cluster 0. This can be explained by the types of documents contained within the two classes. One is Penoscrotal and the other is Andrology. A lot of the same markers are going to be present as they both deal specifically with the male physiology. However, like the bag of words vectors, there are some clusters (3, 4, 5) comprised of a very small number of documents.

Table 23 Clustering against nine classes with Doc2Vec word embeddings

Cluster	UR01	UR02	UR03	UR04	UR05	UR06	UR07	UR08	UR09
0	1.6%	0.9%	0.2%	0.5%	1.1%	1.5%	0.5%	2.2%	1.0%
1	17.9%	16.7%	17.7%	17.3%	15.9%	14.1%	16.2%	16.7%	15.5%
2	11.4%	12.4%	12.5%	13.4%	12.7%	13.0%	12.8%	10.1%	13.5%
3	20.2%	22.1%	22.0%	20.3%	20.8%	22.4%	21.3%	23.6%	19.7%
4	14.9%	15.6%	16.8%	14.0%	13.9%	10.8%	15.9%	11.8%	15.7%
5	10.5%	10.3%	9.3%	11.4%	11.9%	13.5%	10.0%	13.3%	10.1%
6	9.3%	7.3%	8.4%	9.4%	9.0%	9.7%	9.5%	8.6%	9.4%
7	13.7%	13.5%	12.9%	12.6%	14.3%	14.8%	13.1%	13.0%	14.6%
8	0.4%	1.2%	0.2%	1.0%	0.4%	0.3%	0.8%	0.7%	0.5%

The final method of input transformation discussed in regard to the clustering of the urology dataset is Doc2Vec word embeddings. As explained in Section 3.3, The goal of creating word embeddings is to reduce the input data into a manageable context matrix that captures relationships (instead of frequency) between vocabulary members. Unfortunately, the k-means++ algorithm does not interpret the relationships created by the word embeddings in a separable way. Every cluster has members of every class and there are no outliers.

5.2.1.2 Impact of incorporating feature reduction

There are two types of feature reduction applied here to try and improve the clustering on the urology dataset. The first is a step taken before clustering to reduce the size of the input vectors. The original vocabulary size for the urology dataset totalled 66,937 words and phrases when using a range of one to three. By applying PCA, the explained variance ratio, which represents the percentage of variance that is being conveyed by the currently chosen components, was extracted. The explained variance ratio outlined that the number of components needed to represent ninety-five percent of the data was 14,000 terms, which represents a reduction of seventy nine percent.

Table 24 shows the effects of reducing the vocabulary size. The majority of cases are again contained within a small number of clusters with little separation apart from UR04.

Table 24 TF-IDF clustering with a vocabulary of 14,000 terms

Cluster	UR01	UR02	UR03	UR04	UR05	UR06	UR07	UR08	UR09
0	**37.6%**	**81.8%**	**53.0%**	10.8%	17.4%	14.7%	**62.7%**	**50.9%**	**30.9%**
1	0.0%	0.0%	**23.1%**	0.0%	0.0%	0.0%	0.0%	0.0%	13.3%
2	1.1%	1.6%	17.8%	0.8%	0.9%	1.2%	0.8%	1.7%	**43.0%**
3	2.8%	1.9%	0.3%	1.0%	2.9%	3.6%	3.1%	3.7%	1.0%
4	2.3%	3.6%	3.0%	**73.6%**	4.5%	0.5%	6.5%	19.2%	0.0%
5	5.5%	10.6%	1.0%	10.6%	**67.2%**	1.5%	**22.3%**	2.9%	0.3%
6	0.0%	0.0%	0.0%	0.0%	0.6%	0.6%	0.0%	0.0%	1.4%
7	**47.8%**	0.4%	1.2%	0.3%	3.7%	**75.3%**	1.7%	**20.9%**	9.7%
8	2.8%	0.0%	0.7%	2.9%	2.9%	2.7%	2.9%	0.7%	0.3%

The second feature reduction occurs when trying to visualise the data. Using TSNE (van der Maaten & Hinton, 2008) for dimensionality reduction presents an understandable representation of the data in low dimensional space. Figure 5.4 shows 10,000 randomly chosen points within the dataset and through TSNE reduces the dimensions to plot them on a two-dimensional graph. These data points are colour coded to match the cluster assigned. A look at the data may suggest separation between clusters, for example, the blue and red cluster in the top of the graph. When comparing the clusters in Figure 5.3 to the actual classes in Figure 5.4, however, those same clusters are a combination of multiple different classes. The datapoints belonging to the bigger classes in the selection (green, red, blue in Figure 5.4) are represented well in the clusters.

TSNE can have issues representing very high dimensional data in a meaningful way. PCA can be introduced as an initialisation step before TSNE to reduce the data further into a manageable interpretation of the original data. The effect of PCA on the representation can be seen in Figure 5.5 and Figure 5.6. The graphs are using the same k-means clusters as the input data but with a significantly reduced dimensionality, only representing the fifteen most principal components. While the number of components does not represent the total variance of the data within the system, the outputted graphs provide a clear example of the overlap present between classes in the clusters created by the k-means algorithm.

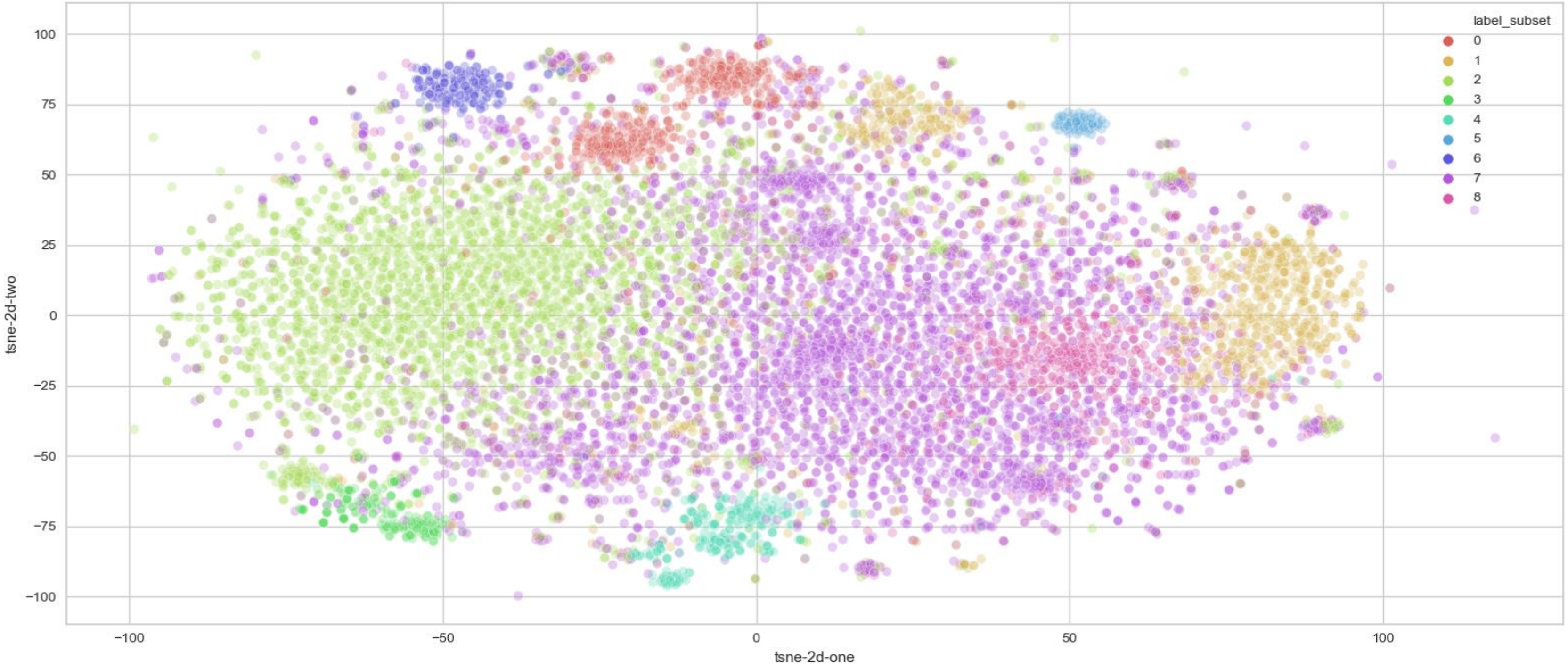


Figure 5.3 TSNE representation of clusters within the urology dataset

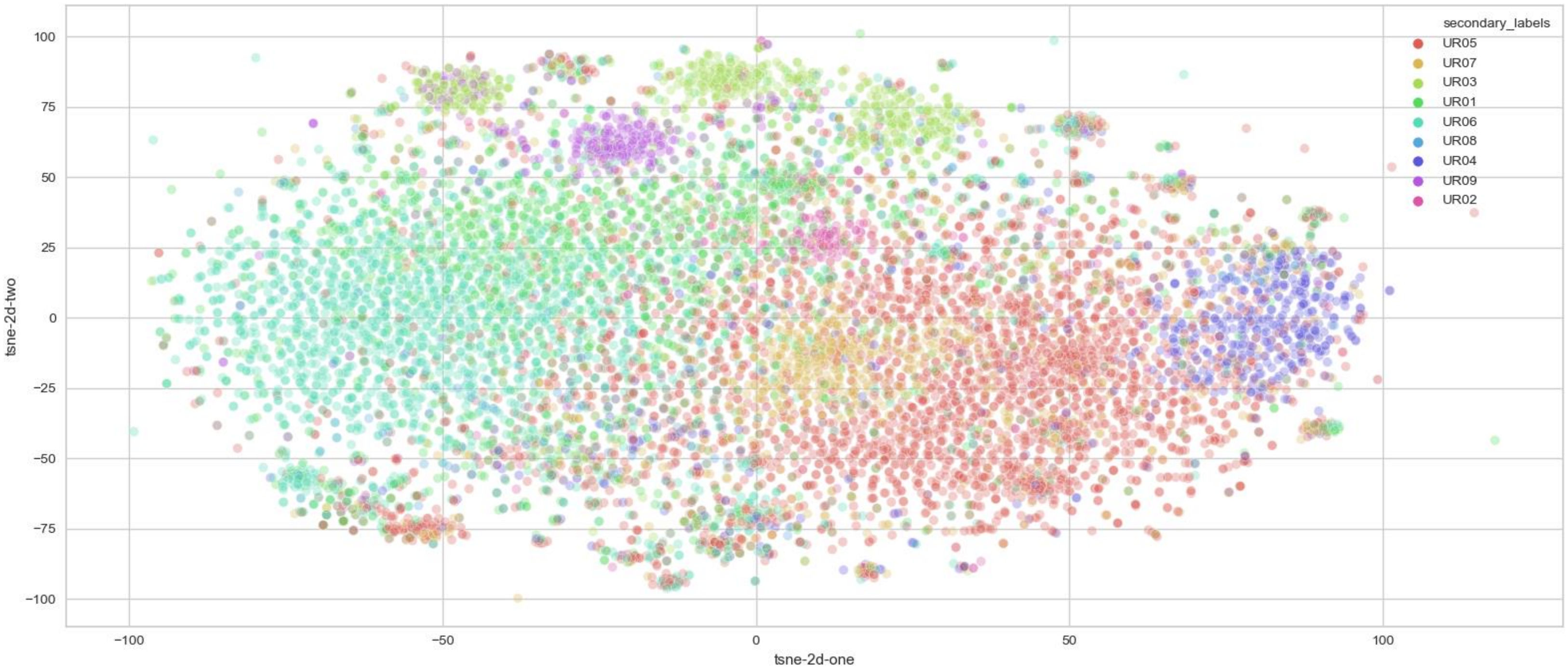


Figure 5.4 TSNE representation of classes within the urology dataset

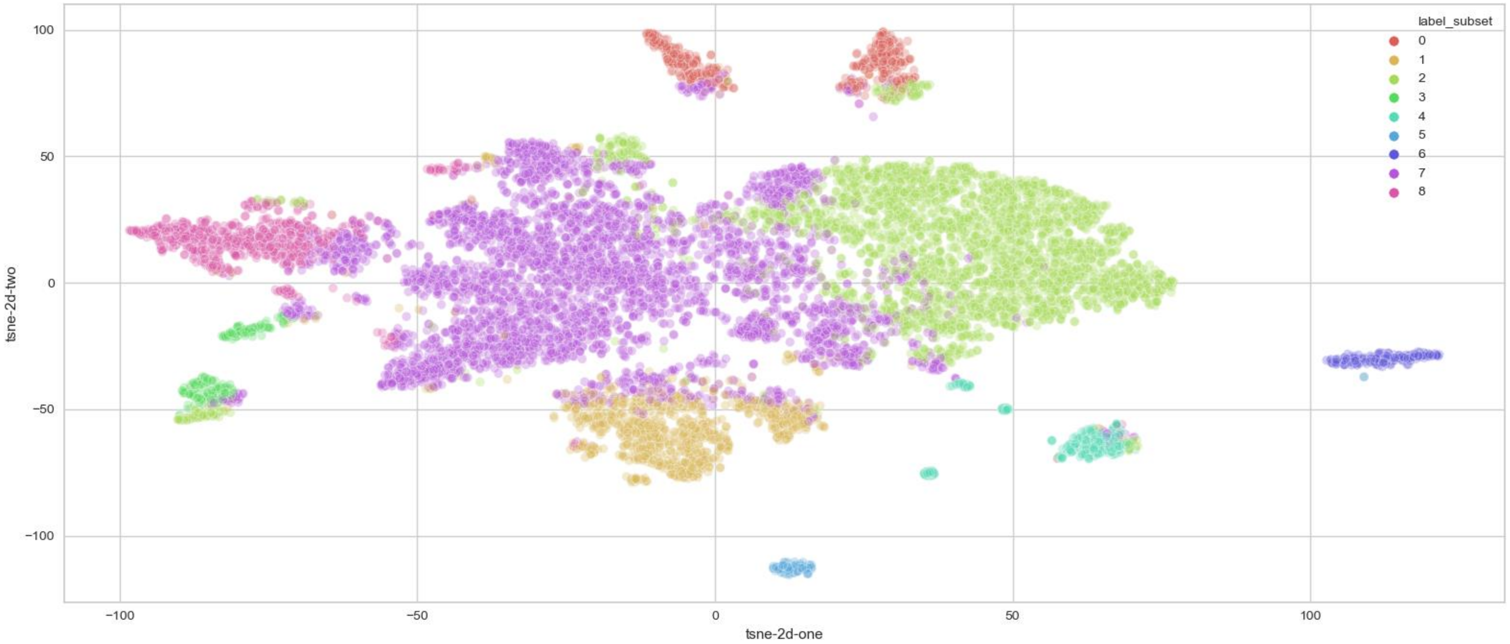


Figure 5.5 TSNE and PCA representation of clusters

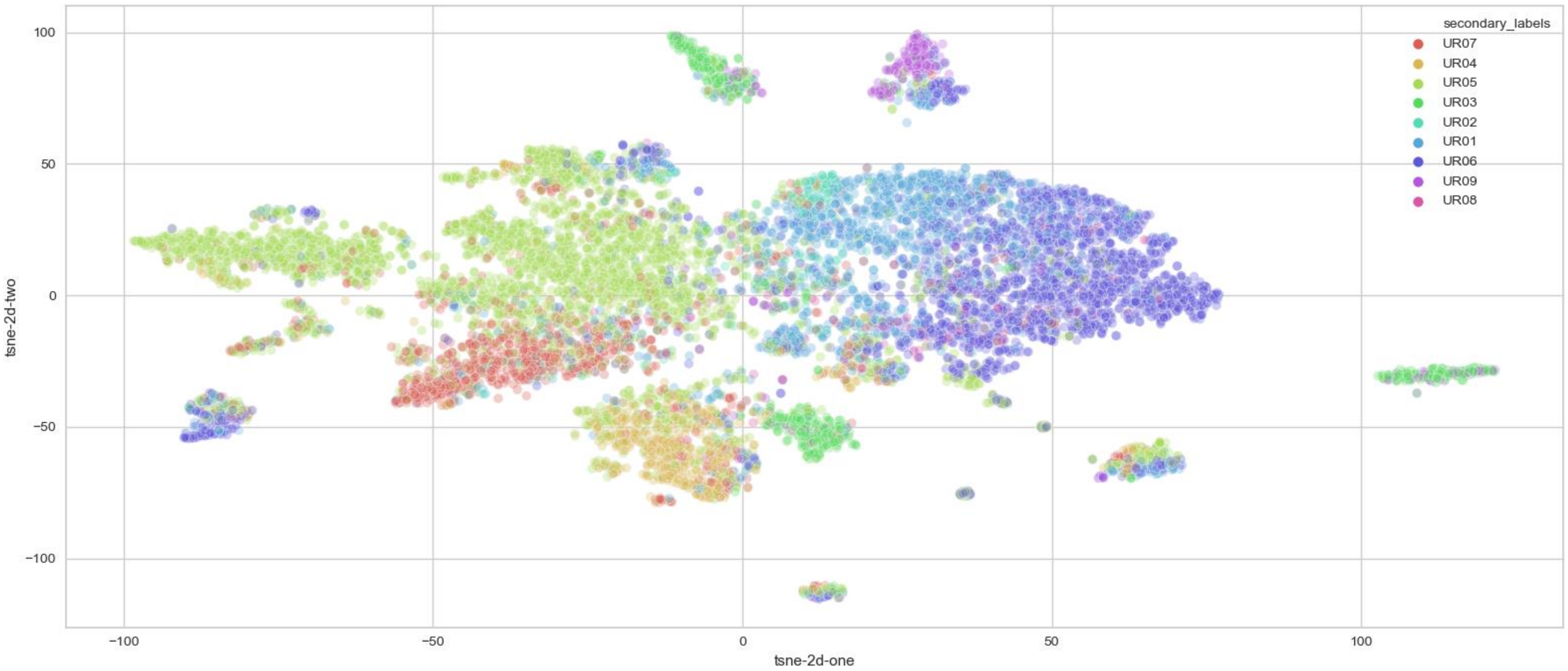
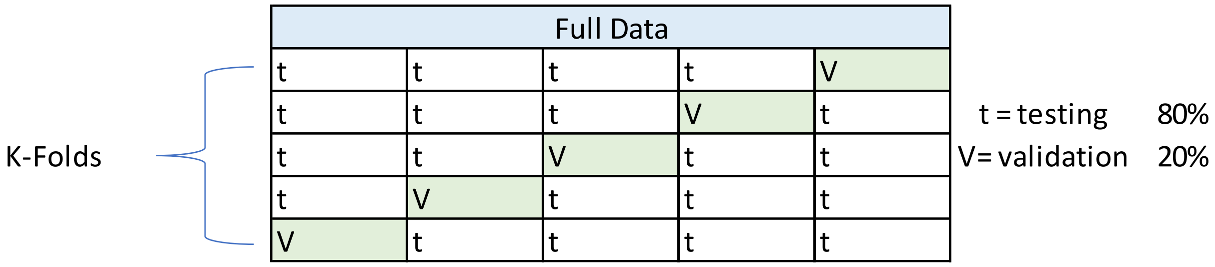


Figure 5.6 TSNE and PCA representation of classes

5.2.2 Classification results

Moving on from clustering, supervised learning techniques for text classification are the next tools to apply to the dataset. Again, the first step is to decide the best vectorisation method for transforming the raw textual data into a workable series of document vectors in which the machine learning algorithms can find and understand the similarities and differences between individual cases. The four methods of vectorisation used with this dataset while looking at supervised learning techniques outside of neural networks are bag of words (frequency counts), one-hot vectorisation, TF-IDF, and Doc2Vec (see Chapter 3). For the supervised learning techniques discussed in this chapter, there are four algorithms tested and compared. A linear kernel support vector machine, logistic regression, multinomial Naïve-Bayes, and a random forest classifier.

To accurately portray the performance of these models, a five-fold cross-validation technique has been used to randomly sample the dataset into testing and validation data. The explanation in Figure 5.7 explains the method behind cross-validation.



Full Data				
t	t	t	t	V
t	t	t	V	t
t	t	V	t	t
t	V	t	t	t
V	t	t	t	t

Figure 5.7 K-Folds cross validation table when K = 5.

The cross-validation method splits the data randomly into K data bins wherein K-1 bins are used for training data and the final bin is used for validation data. This process is repeated K times until each bin has been used for testing purposes once. The results of the K-fold testing are then averaged for the experiments conducted.

Table 24 shows the accuracy achieved by the four different models on the urology dataset alongside the standard deviation. All methods of vectorisation achieve classification results that are similar to one another, with TF-IDF and Doc2Vec presenting the highest accuracies. The reason for no result under the Multinomial Naïve-Bayes with Doc2Vec is that the algorithm does not work with negative values. The alternative would be to scale the input data prior to applying the algorithm but this may result in a reduced level of granularity. Table 25 also shows that from the four model choices, logistic regression and support vector machines are the better choice. While the Naïve-Bayes approach provides a good baseline comparison, the random forest classifier again fails to classify the data like in the presenting complaints dataset (see Table 19).

Table 25 Five-fold cross validation results on Urology dataset

	Count	One-Hot	TF-IDF	Doc2vec
	Accuracy	Accuracy	Accuracy	Accuracy
Linear SVM	78.5%	78.8%	81.2%	83.8%
Logistic Regression	80.5%	80.6%	81.1%	83.6%
Multinomial NB	80.5%	80.3%	70%	NaN
Random Forest	35.7%	35.7%	35.7%	68.5%

However, accuracy does not produce a full overview of a model’s performance on a dataset. Issues that occur when classifying smaller classes can be masked by successes on larger classes. Table 26 and Table 27 show this disparity. While the logistic regression model scored similar or better accuracies with the five-fold cross validation, given a random training and testing set the accuracy is lower than the linear support vector machine. The logistic regression model fails to classify any document related to UR08 at all and does not outperform the SVM on any class in regard to F1-score.

Table 26 Classification Report: Linear SVM on urology

	Precision	Recall	F1-Score	Support
UR01	0.81	0.79	0.80	317
UR02	0.86	0.73	0.79	172
UR03	0.85	0.90	0.87	1112
UR04	0.76	0.69	0.72	515
UR05	0.84	0.90	0.87	355
UR06	0.74	0.40	0.52	141
UR07	0.72	0.73	0.73	718
UR08	0.50	0.15	0.23	74
UR09	0.85	0.91	0.88	1687
accuracy			0.82	5091
macro avg.	0.77	0.69	0.71	5091
weighted avg.	0.81	0.82	0.81	5091

Table 27 Classification Report: Logistic regression on Urology

	Precision	Recall	F1-Score	Support
UR01	0.84	0.74	0.78	317
UR02	0.90	0.65	0.75	172
UR03	0.85	0.90	0.87	1112
UR04	0.80	0.61	0.69	515
UR05	0.85	0.89	0.87	355
UR06	0.96	0.32	0.48	141
UR07	0.71	0.75	0.73	718
UR08	0	0	0	74
UR09	0.8	0.92	0.85	1687
accuracy			0.81	5091
macro avg.	0.74	0.64	0.67	5091
weighted avg.	0.80	0.81	0.80	5091

5.2.2.1 Comparison of classification results between linear and RBF SVM kernels

With the support vector machine model performing the best out of all the other models tested, a comparison was made between the linear kernel and a radial basis function (RBF) kernel. The RBF kernel was tested with different values for the hyperparameters *C* and gamma as they have a greater impact on hyperplane placement when using a non-linear kernel. The best values of these hyperparameters on the urology dataset were found through testing. The value for gamma had a greater impact than the value for *C*, with the best results existing in a range of 0.4 - 0.6. Using a gamma value of 0.5, it was found that the main impact of the *C* parameter was fitting the data to smaller classes like UR08. However, the overall F1-Score for the model was not affected by the *C* parameter unless it was set to one or below. Doing so lowered the accuracy of the overall system, as shown in Table 28.

Table 28 Effects of the *C* hyperparameter on the RBF SVM when Gamma = 0.5

C	F1-Score
1.5	0.81
1	0.80
0.5	0.80
0.2	0.78
0.1	0.75

With the hyperparameters tuned, Table 29 shows the performance of the SVM with an RBF kernel. The RBF kernel and linear kernel perform similarly, with each having classes that produce a higher precision, recall and F1-score than the other.

Table 29 Classification Report: RBF SVM on urology

	Precision	Recall	F1-Score	Support
UR01	0.77	0.85	0.81	317
UR02	0.84	0.71	0.77	172
UR03	0.85	0.89	0.87	1112
UR04	0.72	0.72	0.72	515
UR05	0.83	0.89	0.86	355
UR06	0.83	0.44	0.57	141
UR07	0.68	0.78	0.73	718
UR08	0.65	0.15	0.24	74
UR09	0.88	0.85	0.87	1687
accuracy			0.81	5091
macro avg.	0.78	0.7	0.72	5091
weighted avg.	0.81	0.81	0.81	5091

5.2.2.2 Classification using neural network approaches

After testing the dataset with traditional machine learning models, it was decided to also test the urology dataset using linear and RBF neural networks. All models were built using the Keras (Chollet, 2015) Python library. Many architectures were trialled for the linear neural network and notable ones are shown in Table 30. The linear neural network began to overfit after a single epoch; accuracy would continue to increase on the training set but decrease on the validation (testing) set. To counteract this, dropout layers were added in as per recommendations in Srivastava et al. (2014). A dropout layer randomly selects a portion of the nodes to turn off at each stage of training. The goal of the dropout layer is to reduce the overall amount of the training data the network sees at any given time, reducing the chance of overfitting. A dropout layer of 0.8 was added after the input layer, and dropout layers of 0.5 were added after each hidden layer. While this increases the computational time till convergence, it may improve the generalisation of the model.

Table 30 Linear Neural Network architectures used on urology dataset

Architecture	Validation Accuracy (No dropout)	Validation Accuracy (Dropout)
Input layer: 128 nodes Hidden Layer: 64 nodes Output Layer: 9 nodes	.83 (1 epoch)	.83 (3 epochs)
Input layer: 2048 nodes Hidden Layer(s): 1024 – 512 - 256 - 128 Output Layer: 9 nodes	.82 (2 epochs)	.83 (2 epochs)
Input layer: 9 nodes Hidden Layer: 9 nodes Output Layer: 9 nodes	.81 (1 epoch)	.54 (2 epochs)
Input layer: 256 nodes Hidden Layer(s): 512 - 512 Output Layer: 9 nodes	.82 (1 epoch)	.84 (11 epochs)

Table 31 shows the classification report on the same training data as the models in Section 5.2.2.1. There are improvements to the performance metrics for all classes except UR08, which fell by .05. This overall improvement was made with comparable computation time as the previous models due to the immediate overfitting after a short number of epochs.

Table 31 Classification report: best linear neural network

	Precision	Recall	F1-Score	Support
UR01	0.88	0.76	0.82	317
UR02	0.9	0.69	0.78	172
UR03	0.87	0.89	0.88	1112
UR04	0.77	0.72	0.74	515
UR05	0.82	0.92	0.87	355
UR06	0.8	0.59	0.68	141
UR07	0.75	0.78	0.77	718
UR08	0.57	0.11	0.18	74
UR09	0.86	0.92	0.89	1687
accuracy			0.84	5091
macro avg.	0.8	0.71	0.73	5091
weighted avg.	0.83	0.84	0.83	5091

The same process was repeated for the other types of neural networks. An RBF neural network with a single hidden layer achieved the same accuracy (.81) as the RBF in a support vector machine environment. Two sets of convolutional neural networks were tested, one with a single convolutional layer and a second with four layers to test deep learning. The CNN with a single layer outperformed the CNN with multiple layers with an accuracy of 0.82 compared to 0.79. This may have occurred as all the impactful features could be found in a single layer with no benefit from further separations, just added noise. The final network architecture, the RNN, struggled to produce a comparable accuracy. The best model found had an overall accuracy of 0.79 when employing an LSTM architecture. Unfortunately, while the other networks discussed in this section trained with a computational time of less than a minute for a single epoch, the RNN network averaged thirty-three minutes.

5.2.2.3 Feature reduction techniques applied to the urology letters

The full vocabulary for the urology dataset with the words and phrases is 66,937. This is a large vocabulary for a set of data centred around a small subset of medical knowledge. Taking the current approaches further in Section 5.3 with the general referral letters will significantly expand the vocabulary. This section outlines steps taken to extract a portion of the dataset that still represents the overall relationships.

As previously stated in Section 5.2.1.2, applying PCA to the dataset found that around 14,000 words explained ninety-five percent of the variance present within this dataset. This finding presented an initial target size for any other feature reduction method discussed. The choices made for this feature reduction include the TF-IDF’s max features, employing SVD, a form of recursive feature elimination (RFE) (Guyon, et al., 2002), and a vocabulary based purely on a medical dictionary.

Recursive feature elimination is the process of reducing features in the dataset one step at a time until you reach a chosen maximum number of surviving features. Guyon et al. (2002) outlines these steps for gene selection in a cancer classification problem. After training a support vector machine, features can be ranked according to the weight vectors associated with them. After the features are ranked, the feature with the smallest ranking criterion is removed and the process repeated until the required reduction is completed.

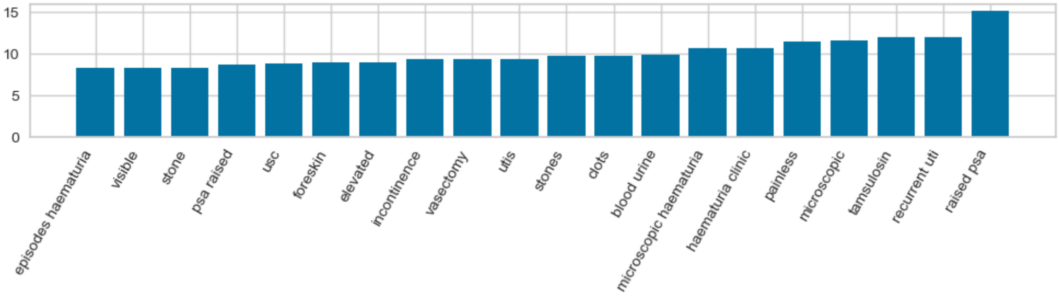


Figure 5.8 Top 20 most influential features on the urology dataset

The approach taken in this thesis is different to the one set out in Guyon et al. (2002). Their approach takes into consideration the positive and negative values for each indicated feature to determine the ranking of a specific feature. In contrast, the original research presented in thesis used the absolute values of features, which then indicates the features that differ the most on average between classes (Furey, et al., 2000) and thus have a greater impact on classification. The results of carrying out the process in this way are shown in Figure 5.8 which shows the top twenty features in the dataset according to the sum of absolute weights associated with them after SVM training. Rather than perform RFE based as per previous literature, the approach taken in this thesis is to instead threshold the data by these absolute weights. Figure 5.9 displays the code written to extract and evaluate the weights present inside the trained SVMs attributes.

```
coef = classifier.coef_
coef = [abs(element) for element in coef]
total_coef = [sum(x) for x in zip(*coef)]
total_coef = np.asarray(total_coef)
feature_names = np.array(tfidfmodel.get_feature_names())
values_to_test = [0.0 … 3.5]
filenames = ['0_0' …  '3_5']
for index, current_val in enumerate(values_to_test):
    indices = [ ]
    for idx, val in enumerate(total_coef):
        if val < current_val:
            indices.append(idx)
    reduced_vocab = delete_from_csr(original_vocab, None, indices)
    current_file = "{}.npz".format(filenames[index])
    scipy.sparse.save_npz(current_file, reduced_vocab)
```

Figure 5.9 Code for extracting features according to coefficient threshold weights

After evaluation, individual sparse matrices were created using the features that existed above the chosen threshold. These sparse matrices were then used as input vocabularies for future experiments. Table 32 shows the effect that the thresholding has on the size of the vocabulary. By excluding the features that present little overall weight to the decision process the size of the vocabulary can be significantly reduced. For instance, removing all the features that have an absolute weight below 1.0 results in a vocabulary size which aligns with the idea of ninety-five percent of the variance in the dataset, 14,106.

Table 32 Urology vocabulary size changes with linear SVM thresholding

Threshold Weight	Vocabulary Size
0.0	55311
0.1	54487
0.2	52254
0.3	48409
0.4	43185
0.5	37422
0.6	31561
0.7	26109
0.8	21445
0.9	17435
1.0	14106
1.1	11484
1.2	9311
1.3	7573
1.4	6211
1.5	5111
1.6	4193
1.7	3446
1.8	2905
1.9	2424
2.0	2046
2.1	1737
2.2	1468
2.3	1271
2.4	1107
2.5	954
2.6	851
2.7	745
2.8	668
2.9	584
3.0	508
3.1	460
3.2	419
3.3	370
3.4	329
3.5	301

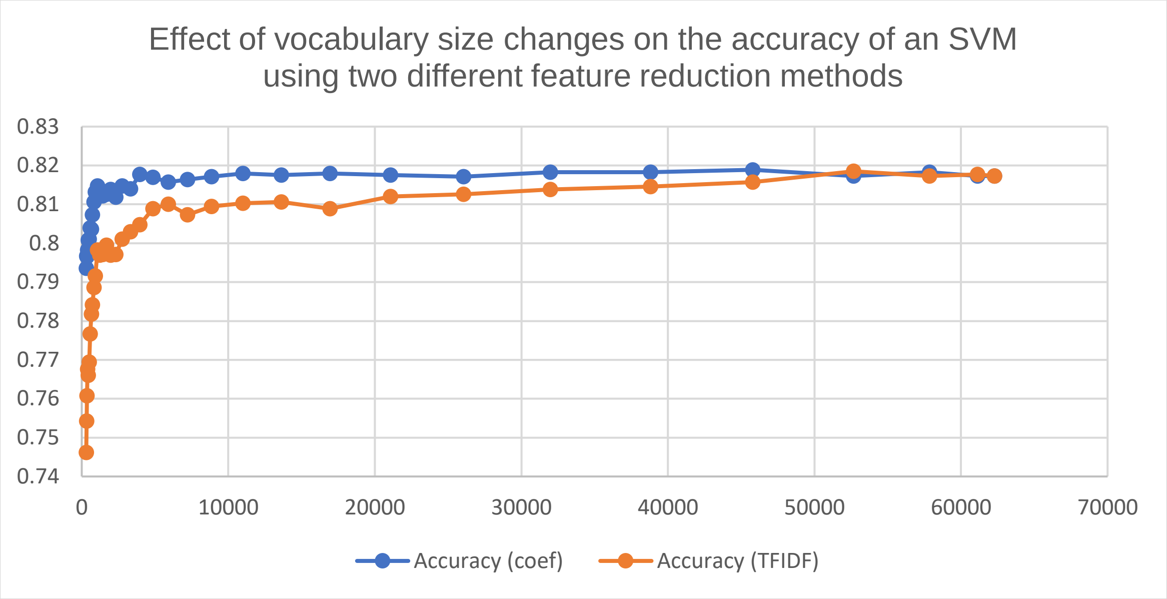


Figure 5.10 Accuracy changes using SVM coefficients and TF-IDF max features

Purely reducing the dataset to better fit computational time and resources is unimportant if the resultant accuracy is worse than the original methods. Figure 5.10 shows the changes in accuracy that occur when reducing the vocabulary size through two of these reduction methods. The accuracy fluctuates until it reaches a small vocabulary size wherein it drops rapidly.

The resultant accuracies of reducing the vocabulary to the size of 14,106 is shown in Table 33. The vocabulary column is the original vocabulary size without any feature reduction. The TF-IDF column is the result employing the max_features parameter with the TF-IDF algorithm. This parameter means that the TF-IDF matrix will only keep the top *n* features after conversion. The SVD column is the result of using SVD on the sparse TF-IDF matrix with the entire vocabulary and the SVM coefficient column is based upon a threshold level of 1.0 after training with the entire vocabulary.

As Table 33 shows, the only method of feature reduction that provides an overall improvement to the representation of the dataset is the thresholded SVM coefficients. It produces the same or better result than using the initial vocabulary on eight out of nine categories. It presents the best result (bolded) in six out of nine categories and overall accuracy and F1-Score. A version of RFE was attempted using Sklearn’s (Pedregosa, et al., 2011) RFE implementation with support vector regression, but it failed to produce a reduced set of features after hours of training.

While the SVD approach presented a better look at UR08 specifically, it runs into an issue with computational time and resources. To reduce the dataset from the initial vocabulary size to the reduced feature set it took thirty-four minutes and used twenty-four gigabytes of RAM to accomplish the task. On a bigger dataset like the one in Section 5.3, the memory overhead would be far larger.

Table 33 Urology data accuracy for different feature reduction methods

Class	Vocabulary	TF-IDF	SVD	SVM Coef.
UR01	**0.80**	0.79	**0.80**	**0.80**
UR02	**0.78**	0.77	0.77	**0.78**
UR03	0.87	0.86	0.87	**0.88**
UR04	**0.72**	0.71	0.71	0.71
UR05	**0.87**	0.85	0.86	**0.87**
UR06	0.52	**0.59**	0.54	0.55
UR07	0.72	0.72	0.72	**0.73**
UR08	0.19	0.19	**0.23**	0.22
UR09	**0.88**	0.87	**0.88**	**0.88**
Model Acc.	**0.82**	0.81	0.81	**0.82**
Model F1	0.80	0.80	**0.81**	**0.81**
Number of most accurate classes	5	1	3	**6**

5.2.2.4 Classifying patient prioritisation from urology letters

While patient prioritisation was not a goal for the urology dataset, experiments have been carried out to allow comparisons between the urology and the general referral letters datasets. The linear and RBF support vector machines have been applied to tackle the prioritisation problem due to their success when classifying sub-specialities. With both GP and consultant priorities listed in the dataset, it allows for four variations in training and testing labels. Experiments have been carried out with each variation, employing the same training and testing set for each. The resulting F1-scores for these experiments are listed in Table 34.

Selecting consultant priorities for testing labels produced the best results. This included using both GP and consultant labels for training. This may suggest that the consultant's priorities are more consistent with the information contained in the letter itself while the general practitioner's priorities are influenced by outside factors. All eight models had issues classifying the USC prioritised notes in comparison to the other two labels. These cases were attributed mostly to the routine label as the precision score for the routine class was lower than the recall and F1-scores meaning that more false positives were attributed to that label.

Table 34 SVMs ability to classify urology prioritisation

Labels	Prioritisation class	Linear SVM	RBF SVM
GP train to GP output	Routine	0.78	0.78
	Urgent	0.88	0.88
	USC	0.55	0.58
	Overall	0.80	0.80
GP train to consultant output	Routine	0.80	0.80
	Urgent	0.89	0.88
	USC	0.58	0.60
	Overall	0.81	0.81
Consultant train to GP output	Routine	0.80	0.79
	Urgent	0.89	0.88
	USC	0.58	0.61
	Overall	0.81	0.81
Consultant train to consultant output	Routine	0.83	0.82
	Urgent	0.90	0.90
	USC	0.64	0.65
	Overall	0.84	0.83

5.2.3 Observations

Experimentation with the urology dataset supported the premise that there would be significant overlap between the classes. The clustering results in Section 5.2.1 showed that even though a combination of TF-IDF vectorisation and k-means++ clustering produced some measure of separability between the classes, certain classes shared locations in Euclidean space. This was due to symptoms present in classes like UR03 (penoscrotal) and UR09 (andrology) being highly linked.

Classification of specialities was successfully carried out to an accuracy of eighty three percent despite these overlaps, with the white box SVM and black box neural network producing the best results. The results indicate that with a wider, more separable, set of features in the generalised referral dataset (see Section 5.3), the same models should be able to classify the data with a higher degree of accuracy. Inversely, the random forest classifier used in this research produced results that could not be considered usable. This is believed to be because the random forest classifier can have issues with high dimensional or sparse datasets as mentioned in the usage documentation for the sci-kit learn library (Pedregosa, et al., 2011). This issue has been seen in prior research where Karlsson & Boström (2014) found that when using a sparse medical dataset to predict for adverse drug effects, the random forest algorithm was heavily biased towards the majority class which presented a low overall performance score. This would also indicate why the algorithm performed better with the Doc2Vec inputs as they are a dense matrix of relationships rather than a sparse frequency-based matrix.

5.3 Results gathered from the general referral dataset

The urology dataset provided an idea of how classification and clustering techniques can perform using full-bodied medical letters. The methods applied produced results with accuracies above eighty percent and presented a way of reducing the datasets vocabulary to one that is smaller but still representative.

The goal of applying the same Natural Language Processing techniques to this referral dataset is then to determine whether a selection of full-bodied letters presenting a wider spectrum of specialities are inherently more categorizable. This goal comes from the idea that the differences in symptoms for medical conditions are more apparent than those in the urology sub-specialties. The steps taken in the following sections were based upon the information gathered from the work conducted with the urology and presenting complaints data.

5.3.1 Clustering results for the general referral letters dataset

The first stage involved implementing the k-means clustering method and seeing if the algorithm could associate clusters with individual or groups of specialties. The expectation behind this method was that the cases about specialties like urology should be distinctly different and identifiable from other cases assigned to dermatology. With cluster analysis being an unsupervised method there is the expectation of overlap between certain classes in this dataset that belong to the same genre of speciality (community orthopaedic and orthopaedic as an example) but fewer overlaps than those present within the urology dataset discussed in Section 4.2.2. The unbalanced nature of the dataset will also have a significant impact on the ability of the algorithm to cluster cases correctly. Without the knowledge of which class each case belongs to, the algorithm cannot adjust towards a more correct result. To compensate for this, clustering is used to confirm the basis of the belief that the dataset is separable.

While k-means as an algorithm scales well to many datapoints, it struggles with the curse of dimensionality. Converting all 111,128 cases to a TF-IDF vector matrix using an n-gram range of 1, 5 results in a vocabulary size of 306,936. This affects the clustering algorithm in multiple ways. The substantial number of dimensions cause an exponential increase in computing power and time needed to cluster the cases, and more importantly, it makes individual cases more likely to converge together. This convergence occurs because although the vocabulary size of the dataset is 306,936, the cases exist in sizes of 10-251 and some words within a letter may repeat. This results in a sparse document vector, where many of the dimensions will have a value of 0 and as such most cases will be inseparable for these dimensions.

Figure 5.11 shows the results of running the k-means elbow visualiser from the Yellowbrick (Bengfort & Bilbro, 2019) Python library on the data for a range of clusters between 2 and 29 (the number of classes in the dataset) against a distortion score. The alternative method for scoring clusters (the Calinski-Harabasz index score) is unsuitable for this larger dataset as it requires a dense dataset instead of a sparse dataset, which requires a larger memory profile than was available for this project.

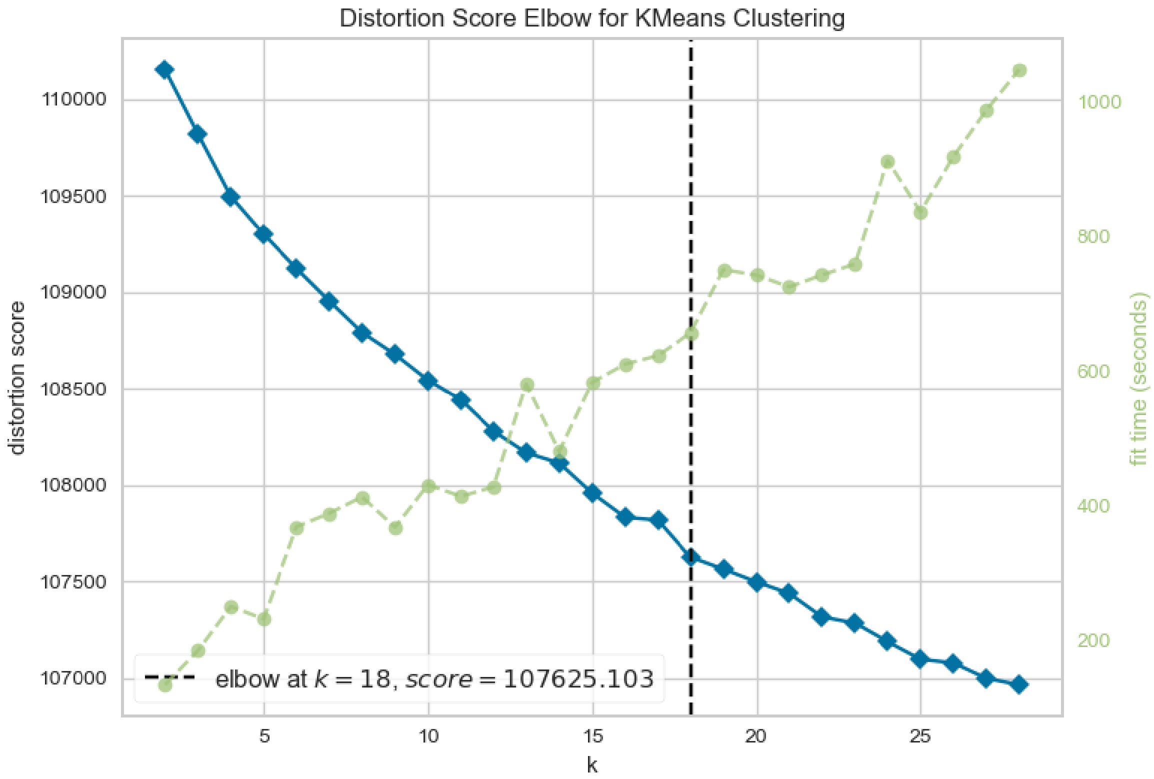


Figure 5.11 Distortion score for clusters on the NHS Wales referral dataset

As shown in the graph, the system predicts that the best value for the clusters is eighteen according to the distortion score. This result is expected due to the reasons stated above: crossovers within the classes and the sparse nature of the input data. Following the same approach as the urology dataset in 4.2.2, the choice was made to set the number of clusters (K) to the number of existing classes (K^I).

With a high SSE of 106859.34 when looking at 29 clusters, the next stage is to consider dimensionality reduction through PCA (Principal Component Analysis) or truncated singular value decomposition (SVD). Implementing the PCA function from the scikit-learn library allows for the linear reduction in dimensionality through Singular Value Decomposition, however the method does not work with sparse data. To convert the data to a dense format requires a large amount of memory and while this was acceptable for the smaller urology dataset this larger dataset requires more memory than is available (251GB). The truncated SVD method is suitable for sparse data and is used for the dimensionality reduction that follows.

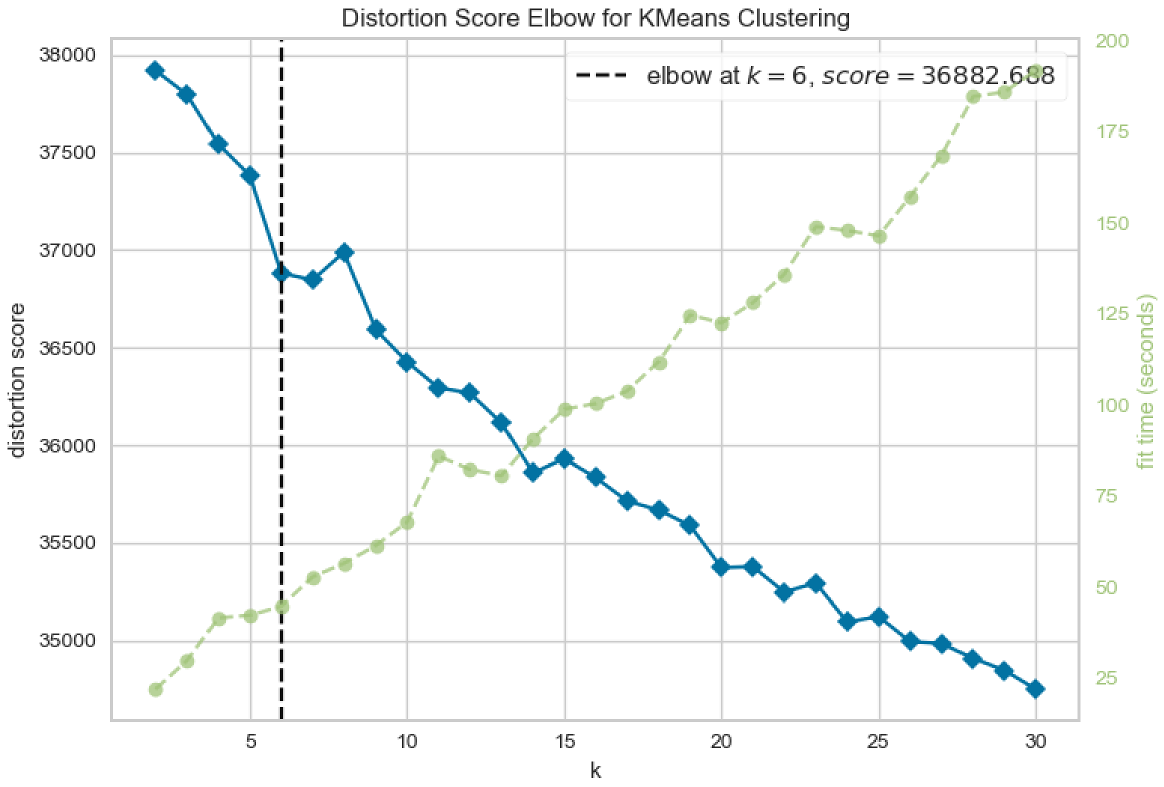


Figure 5.12 Distortion score for clusters on the SVD reduced referral dataset

After reducing the dataset through truncated singular value decomposition down to one thousand relevant features, re-running the k-means clustering algorithm with a target of twenty-nine clusters gives us a new SSE of 34835.33. The reduction in vocabulary results in a significantly faster completion than the earlier runs with the full vocabulary of features. With the reduced number of features, the size of the data is now of a suitable size to use the Calinski-Harabasz score. Figure 5.13 shows that the metric predicts that the variance within the data degrades significantly as the number of clusters increases.

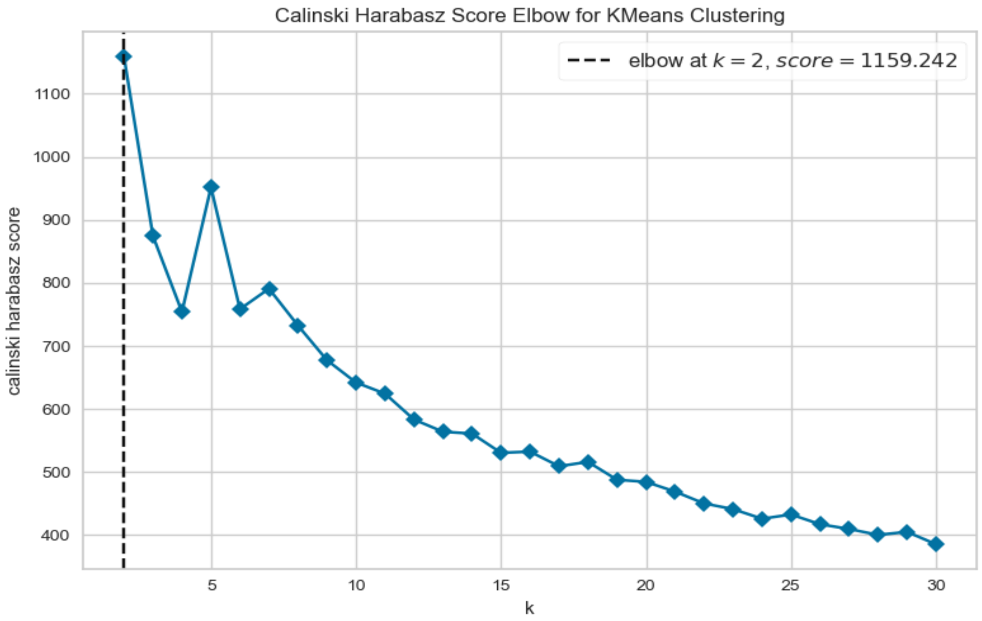


Figure 5.13 Calinski-Harabasz score for the SVD reduced referral data.

The predicted clusters can be visualised using t-distributed stochastic neighbour embedding (van der Maaten & Hinton, 2008), allowing for the conversion and placement of the high dimensional data onto a two-dimensional plane. As there is 111,128 datapoints within the dataset, the following graphs are a randomized subplot of 10,000 examples. Figure 5.14 shows the full dataset plotted onto a two-dimensional plane, with each colour depicting a predicted cluster. Figure 5.15 shows the same data plotted in the same way; except this time the colours stand for the actual speciality class that the case belongs to.

Figure 5.14 and Figure 5.15 show that while several clusters appear to be separable, those clusters are made up of a variety of different classes. For instance, cluster fifteen on the far left of the graph, while made up of several classes, appears to be a densely packed cluster of cases separated from the rest of the examples. This is a downside to tSNE in which distance between clusters is not actually indicative of anything important. The data presented in Figure 5.14 and Figure 5.15 also show that whilst the k-means algorithm manages to cluster large amounts of the same class together, using general surgery as an example, the same class can dominate one cluster but be a large part of several other clusters. Using the data present in Appendix 3, we can see that dermatology is dominant in four different clusters: 28, 23, 8 and 5. It also has a strong presence in cluster one where the k-means algorithm has grouped more than a third of all the existing cases.

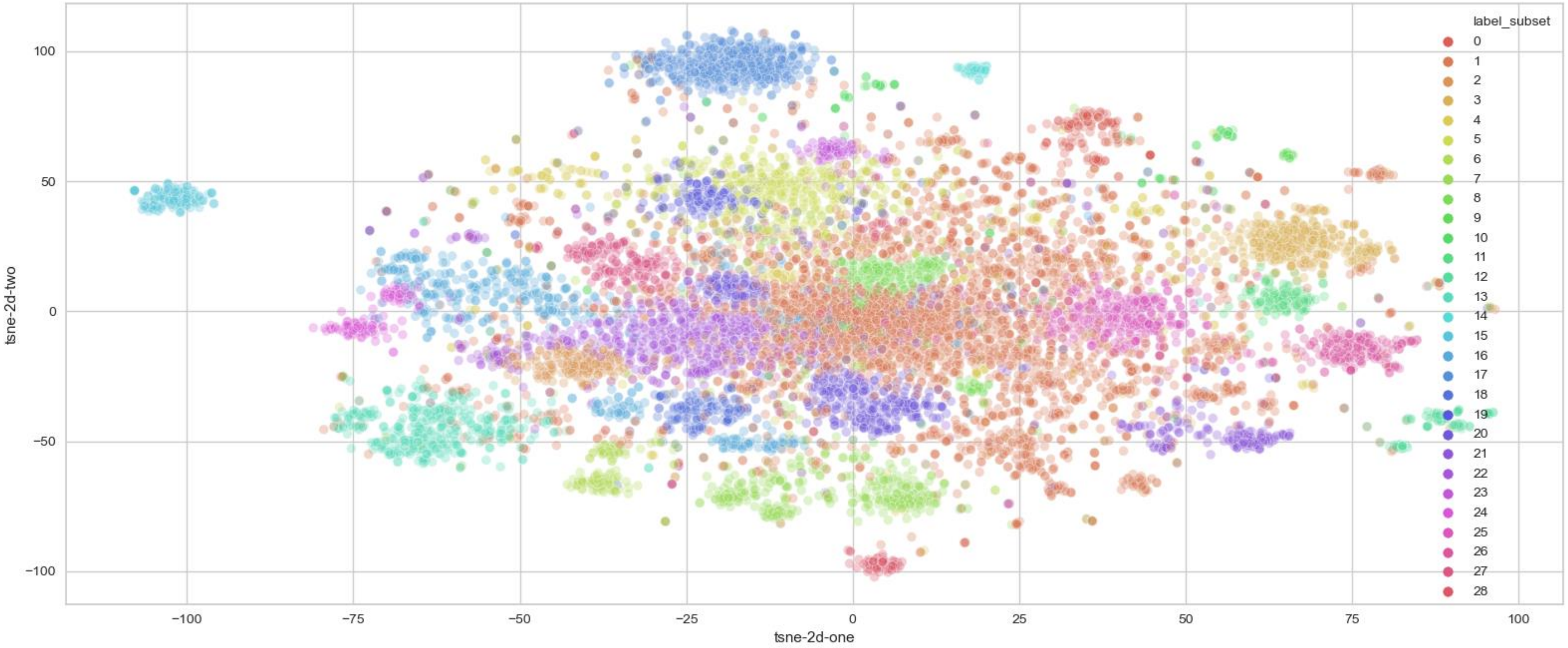


Figure 5.14 tSNE plot of 10,000 examples without feature reduction coloured to the predicted clusters

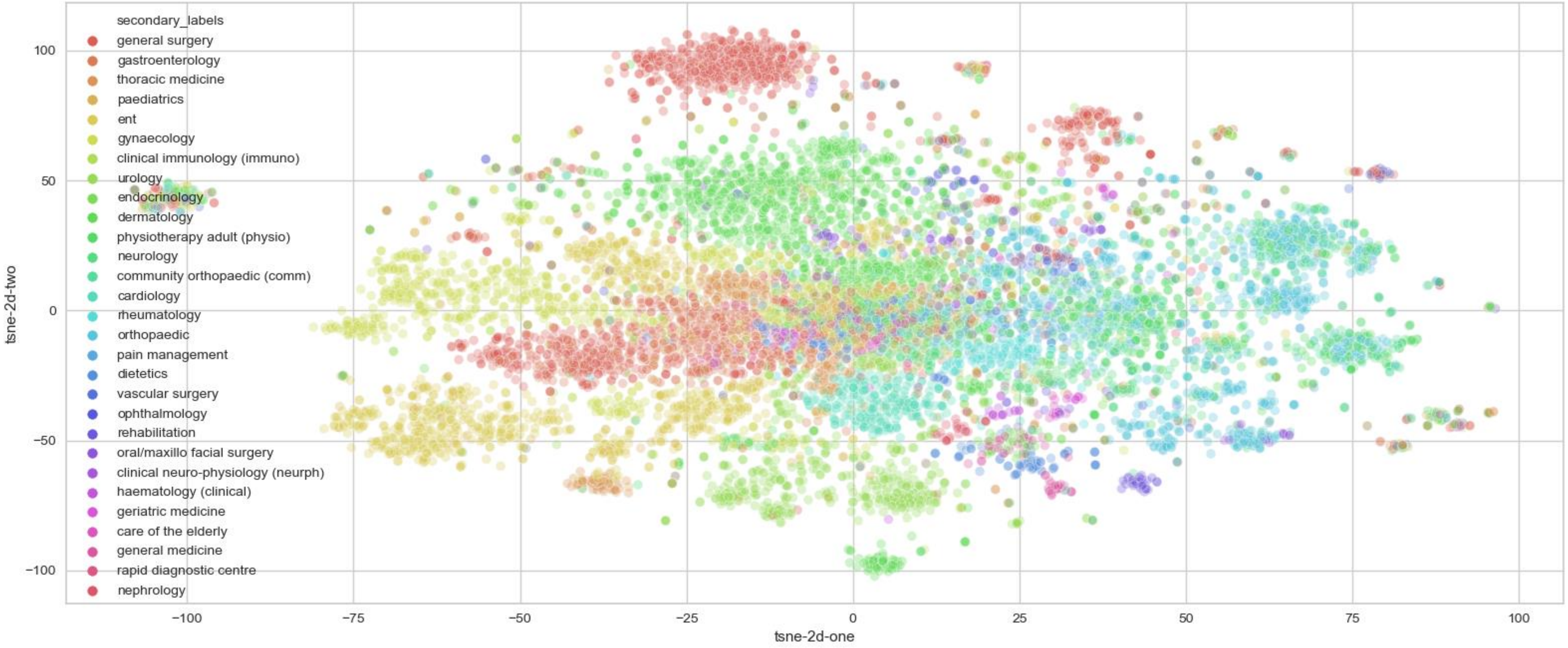


Figure 5.15 tSNE plot of 10,000 examples without feature reduction coloured to actual classes within the data.

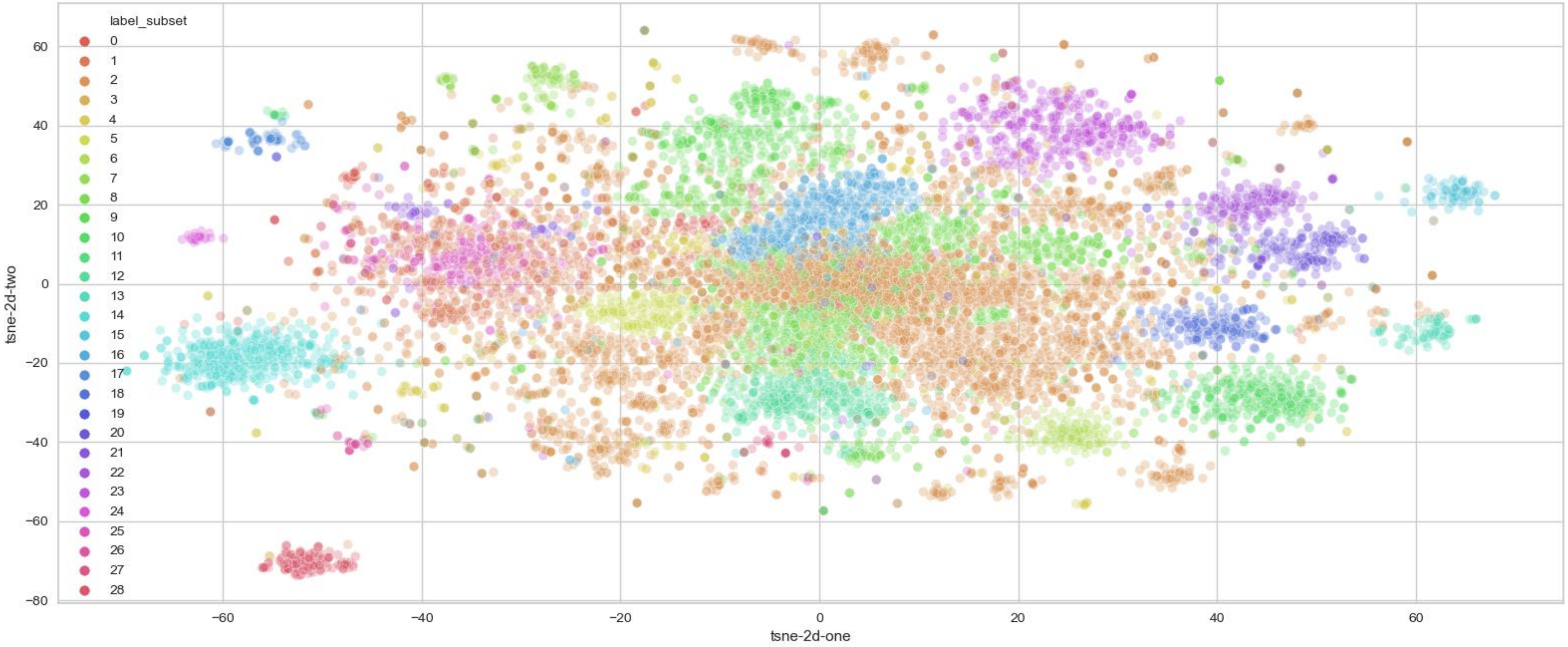


Figure 5.16 tSNE subplot of clusters with a vocabulary feature size of 2500

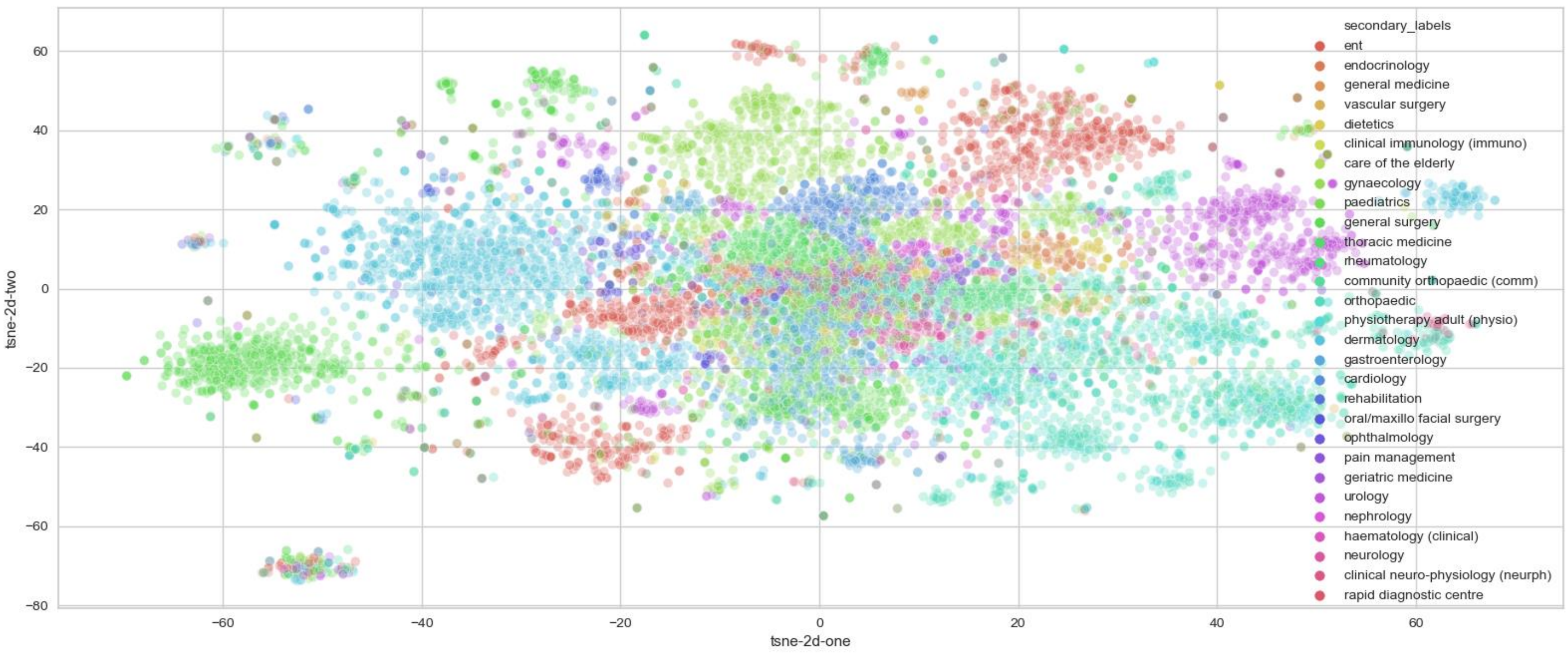


Figure 5.17 tSNE subplot of classes with a vocabulary feature size of 2500

Figure 5.16 and Figure 5.17, alongside the data present in Appendix 4, show that despite the feature reduction the clusters mirror those present in the unmodified dataset. This mirroring shows that while there is a large number of words and phrases that construct the overall vocabulary, there are only a small portion of features that have a significant impact on the separation between documents. This means that employing a feature reduction method before classifying the data should benefit the resultant model by removing unnecessary, noisy features that would otherwise blur the boundaries between classes.

The clusters pictured match to certain classes within the dataset, usually those classes that have a larger number of examples. However, the issue of one large cluster is once again present, and clusters such as cluster zero in this example are a bucket consisting of a wide variety of classes with no clear indicator towards a specific label. Using dermatology again as the example class and excluding the largest cluster (cluster five), the reduced dataset has only created one major dermatology cluster although small numbers of examples are spread across several other clusters.

The results of clustering show that there are differences between the words present within the data for the referral letters and although the algorithm was unable to separate for the substantial number of classes (twenty-nine) present for individual specialties, clustering on this dataset may be possible for target labels of smaller sizes such as assigning patient priorities. The results also show the potential for using supervised learning, the ability to tune a model to the pre-existing labels should provide the opportunity to help separate the inconsistencies and overlapping features present within the data. The following section focuses on exploiting that fact to classify the data using both white box and black box techniques.

5.3.2 Classification results for the general referral letters dataset

The classification conducted on the general referrals’ dataset follows the same methodology that was listed in 5.2.2. The focus will be on mapping the data to twenty-nine specific specialities using a range of classification techniques. The process again looks at various kinds of vectorisation and their impact on the accuracy of the models versus the choice in model itself before moving on to feature reduction methods. This section then compares the ability of classifying specialities with a large number of classes to the three patient prioritisations.

5.3.2.1 Impact of vectorisation choice when classifying

The effect that vectorisation has on a model's ability to successfully classifying data already been shown in 5.2.2. To ensure that the experiments reflect on the vectorisation method instead of the models themselves, the hyperparameters for the machine learning techniques were standardised across the experiments using the bag of words method, TF-IDF and Doc2Vec (See Section 3.3).

Using the six classifiers set up as shown in Table 18, 5-fold cross validation is used to measure the accuracy and F1-Score of each classifier. Table 35 and Figure 5.18 show that when the bag of words vectorisation technique is used with scoring goal of accuracy (correct classifications over all cases), the outcome is three high performing classifiers in Logistic regression, linear support vector machine and stochastic gradient descent. While the random forest classifier achieves comparable results with the two naïve-bayes techniques, the results are significantly lower than the accuracies achieved by the first three techniques.

Table 35 Accuracy with bag of words vectorisation for six classification models

	N-Gram Range							
	1, 1	1, 2	1, 3	1, 4	1, 5	2, 5	3, 5	4, 5
Linear Support Vector Machine	89.129	90.967	91.077	**91.120**	91.116	88.227	76.464	52.340
Logistic Regression	90.432	**91.632**	91.624	91.630	91.623	88.738	77.111	50.632
Stochastic Gradient Descent	88.776	91.076	91.180	91.207	**91.208**	87.633	77.800	49.165
Random Forest	83.785	84.655	84.539	84.331	84.231	84.334	72.719	47.825
Bernoulli Naïve-Bayes	84.731	82.009	81.300	81.170	81.043	63.536	40.796	24.741
Multinomial Naïve-Bayes	80.506	76.116	74.434	73.658	73.226	67.543	54.182	37.712

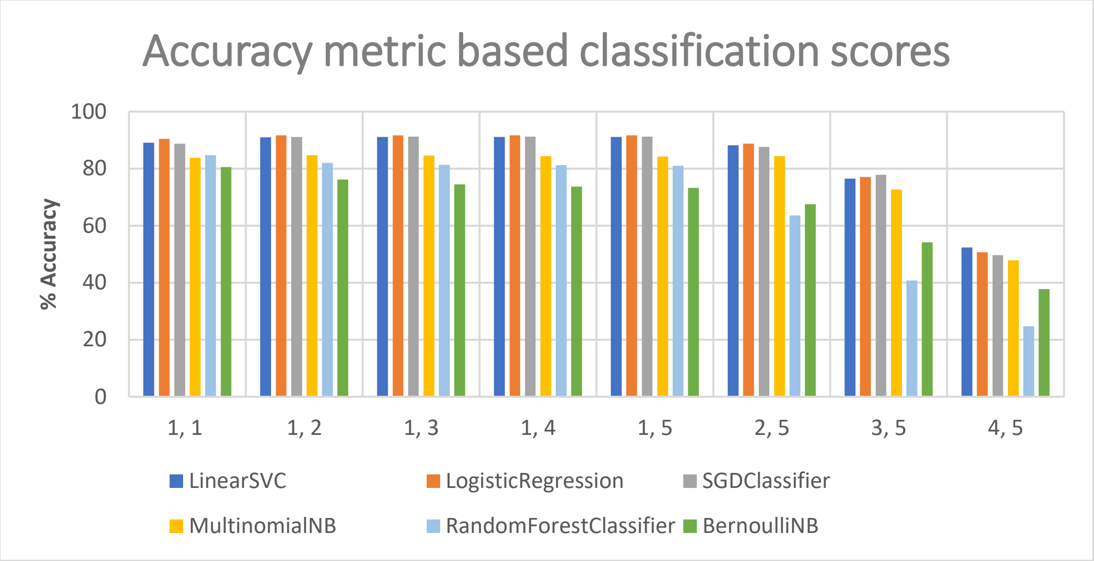


Figure 5.18 Accuracy of classifiers using the bag of words vectorisation technique

Also found in Table 35, the range of n-grams influences the accuracy of the supervised learning models. Figure 5.19 shows that an achievable increase in accuracy when extending the n-grams beyond single words to include phrases for models such as logistic regression and support vector machines. However, the random forest and the Bernoulli Naïve-Bayes classifiers struggle to classify the data when expanding n-gram ranges to include phrases.

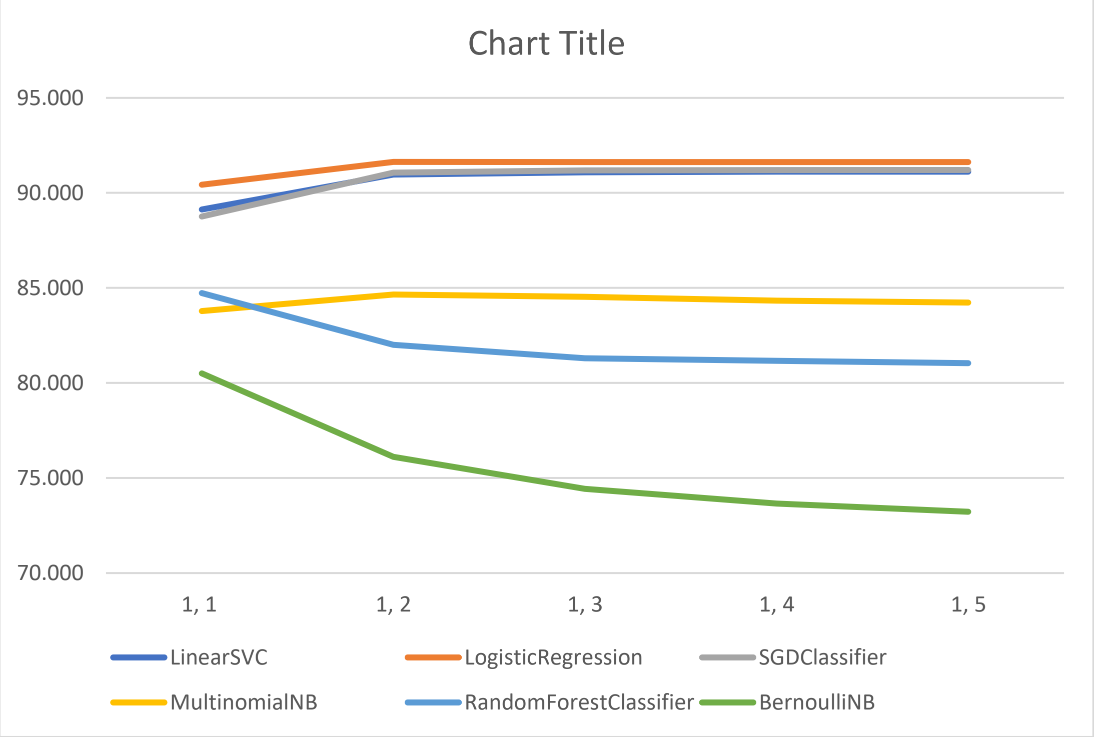


Figure 5.19 Accuracy changes against n-gram range with bag of words

Table 36 F1-Score for bag of words vectorisation on six classification models

	N-Gram Range							
	1, 1	1, 2	1, 3	1, 4	1, 5	2, 5	3, 5	4, 5
Linear Support Vector Machine	89.101	90.893	90.996	**91.039**	91.034	88.074	76.304	53.422
Logistic Regression	90.368	**91.547**	91.532	91.537	91.532	88.521	76.720	50.862
Stochastic Gradient Descent	88.607	91.021	91.085	91.043	**91.098**	87.318	77.315	50.404
Random Forest	83.774	80.248	79.423	79.191	79.099	60.264	40.422	21.821
Bernoulli Naïve-Bayes	79.555	71.692	69.560	68.666	68.203	61.740	50.555	36.137
Multinomial Naïve-Bayes	83.750	83.655	83.451	83.189	83.066	83.759	71.638	47.091

The F1-Score is a better metric than accuracy to use to correctly identify the efficacy of the models. This is due to the GP referral dataset is imbalanced, with some classes having 2/125th of the cases than the highest class and that preventing false positives and false negatives are more important than just the true positives due to the cost of wasting clinician time if referrals are sent to the wrong specialist. When comparing the results in Table 35 and Table 36, the same trend in models is observed. Logistic regression, support vector machine and stochastic gradient descent perform significantly better than the other three models. Whilst the results are marginally lower than those seen when using accuracy as a metric (on average 0.100375 lower for logistic regression), the scores suggest that the classifiers are able to model the relationships in the data when using bag of words vectorisation. Figure 5.20 displays the results of Table 36 in a comparable format. The results shown echo Figure 5.19, the same trends exist where three of the models classify data with an F1-Score of above ninety percent for all n-gram ranges that include individual words, but no model classifies to that level when only including phrases.

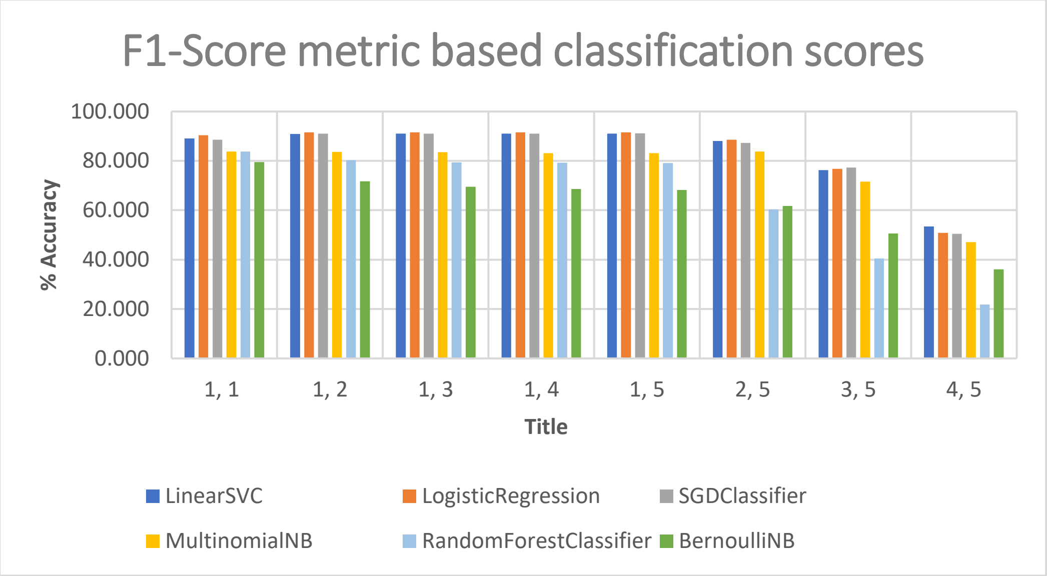


Figure 5.20 F1-Score using the bag of words vectorisation technique

The second method of vectorisation is using TF-IDF to produce weighted scores according to the rarity in which vocabulary terms appear in the dataset (see Section 3.2.2). The same 5-fold cross validation technique is used alongside the same six models and their associated hyperparameters. The accuracy of these models can be seen in Table 37.

Table 37 Accuracy with TF-IDF vectorisation for six classification models

	N-Gram Range							
	1, 1	1, 2	1, 3	1, 4	1, 5	2, 5	3, 5	4, 5
Linear Support Vector Machine	90.106	**91.989**	91.954	91.928	91.914	89.523	79.148	52.415
Logistic Regression	87.763	**88.323**	88.306	88.275	88.260	83.357	71.314	48.477
Stochastic Gradient Descent	85.442	**85.828**	85.733	85.705	85.690	83.132	76.056	48.863
Random Forest	84.960	82.369	81.787	81.340	81.303	63.514	40.695	24.707
Bernoulli Naïve-Bayes	80.506	76.116	74.434	73.658	73.226	67.544	54.182	37.712
Multinomial Naïve-Bayes	78.543	73.746	73.612	73.414	73.331	74.665	66.737	46.305

In comparison to the accuracy scores given under the bag of words vectorisation, most models achieve a lower overall accuracy score. However, the support vector machine approach achieves a greater peak than any of the models previously shown. Accuracy is an imperative part of this thesis due to the field of study (medical) and as such these improvements should not be overlooked. Figure 5.21 again shows the effect that changing the range of n-grams has on the six models' ability to classify the data to the medical specialties. The key difference between the accuracies in the bag of words and TF-IDF vectorisation approaches is that the logistic regression and SGD classifiers never reach the same level of accuracy as the support vector machine, no matter the range used.

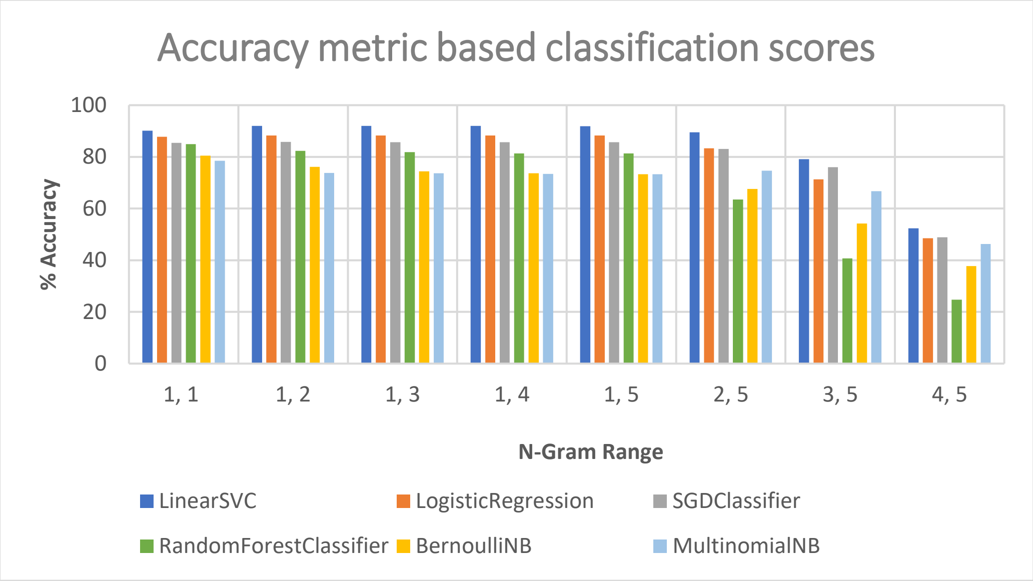


Figure 5.21 Accuracy of classifiers using the TF-IDF vectorisation technique

The F1-Scores for the same models have been provided in Table 38. The same pattern appears in regard to the reduction in scores. The scores of the Bayesian approaches suffer most with an average of 3.9 and 4.3 percent loss in overall score. Comparatively the SVM technique changes the least, with an average of 0.1 loss to performance. This change in performance indicates that the SVM produces a better outlook of all of the classes despite their size whereas the classifications by Bayesian approaches have congregated towards the larger classes.

Table 38 F1-Score for TF-IDF vectorisation on six classification models

	N-Gram Range							
	1, 1	1, 2	1, 3	1, 4	1, 5	2, 5	3, 5	4, 5
Linear Support Vector Machine	89.927	91.854	91.813	91.786	91.177	89.320	78.857	53.256
Logistic Regression	87.314	87.772	87.727	87.691	87.675	81.917	68.755	47.178
Stochastic Gradient Descent	84.280	84.656	84.541	84.540	84.525	81.978	75.048	51.207
Random Forest	83.955	80.879	79.608	79.185	78.957	63.896	53.079	42.467
Bernoulli Naïve-Bayes	79.470	72.399	70.546	69.675	69.110	64.493	58.413	54.521
Multinomial Naïve-Bayes	75.067	68.437	68.216	67.950	67.851	69.966	61.961	46.305

The word embedding matrix approach was also considered for use with the six classification models shown in this section. Doc2Vec matrices were employed on the first four models, with the Bayesian models excluded due to the issues regarding negative feature weights. Two instances of the doc2vec matrices were tested, one that was trained with a small number of epoch's (20) and one with a high number of epoch's (10,000).

Table 39 Doc2Vec accuracies on referral dataset

	Low Epoch	High Epoch
Linear SVM	0.24264	0.14798
Logistic Regression	0.24159	0.34825
Random Forest	0.15758	0.21376

The resulting results of these two Doc2Vec models are shown in Table 39 which are based on an n-gram range of 1, 2. Training the embedding matrix for a significantly longer time did not have a significant impact on the accuracy of models learning from it, with the linear SVM reporting a lower accuracy with the model that was trained for longer.

5.3.2.1.1 Classification report comparison for the best white-box techniques

As seen with the urology dataset, relying solely on using cross validation can mask some issues that a model has with the classification of individual classes. Table 40 and Table 41 compare the ability of the linear SVM and logistic regression when given the same seeded training and testing set for the referral data when using TF-IDF vectorisation.

Table 40 Classification Report: Linear SVM on referral dataset

	precision	recall	f1-score	support
Cardiology	0.95	0.97	0.96	682
Care of the elderly	0.93	0.66	0.77	59
Clinical Immunology	0.91	0.81	0.86	88
Clinical neuro-physiology	0.87	0.75	0.80	44
Community Orthopaedic	0.73	0.53	0.62	393
Dermatology	0.96	0.97	0.97	3017
Dietetics	0.97	0.93	0.95	346
endocrinology	0.85	0.82	0.83	201
ENT	0.95	0.98	0.96	2551
Gastroenterology	0.83	0.88	0.85	987
General medicine	0.84	0.68	0.75	202
General Surgery	0.93	0.92	0.93	2875
Haematology (clinical)	0.94	0.89	0.92	146
Nephrology	0.97	0.87	0.92	87
Ophthalmology	0.90	0.88	0.89	126
Oral/Maxillo facial surgery	0.91	0.79	0.84	218
Orthopaedic	0.86	0.93	0.89	2327
Paediatrics	0.91	0.78	0.84	741
Pain Management	0.88	0.78	0.83	147
Physiotherapy	0.87	0.87	0.87	1348
Rapid diagnostic centre	0.93	0.45	0.61	31
Rehabilitation	0.91	0.90	0.91	128
Rheumatology	0.91	0.92	0.91	430
Thoracic medicine	0.94	0.97	0.96	594
Urology	0.95	0.98	0.97	1766
Vascular surgery	0.82	0.86	0.84	146
Gynaecology	0.97	0.97	0.97	2024
Neurology	0.90	0.87	0.89	441
Geriatric medicine	0.88	0.79	0.83	81
Accuracy			0.92	22226
Macro average	0.90	0.84	0.87	22226
Weighted average	0.92	0.92	0.92	22226

From the overall accuracies presented already, it was known that the SVM would produce the better overall accuracy. However, it performs significantly better at classifying the smaller classes present within the dataset. Using a specific example, the rapid diagnostic centre is a class that represents a Welsh NHS service for referring patients for screenings of potential cancer. These screens cover a wide range of specialties, which means they will have overlapping features with the other classes. The SVM manages to achieve a sixty one percent F1-Score despite the issues with sharing features. The logistic regression model in comparison achieves a precision, recall and F1-Score of zero. This is reflected in the difference between the weighted average and macro averaged F1-Scores for each model’s classification report.

Table 41 Classification Report: Logistic regression on referral dataset

	precision	recall	f1-score	support
Cardiology	0.92	0.96	0.94	682
Care of the elderly	0.76	0.32	0.45	59
Clinical Immunology	0.95	0.6	0.74	88
Clinical neuro-physiology	0.81	0.5	0.62	44
Community Orthopaedic	0.73	0.3	0.43	393
Dermatology	0.93	0.98	0.95	3017
Dietetics	0.96	0.87	0.91	346
endocrinology	0.82	0.72	0.77	201
ENT	0.91	0.97	0.94	2551
Gastroenterology	0.77	0.84	0.80	987
General medicine	0.83	0.52	0.64	202
General Surgery	0.89	0.91	0.90	2875
Haematology (clinical)	0.93	0.71	0.80	146
Nephrology	0.98	0.67	0.79	87
Ophthalmology	0.92	0.68	0.79	126
Oral/Maxillo facial surgery	0.91	0.58	0.71	218
Orthopaedic	0.79	0.92	0.85	2327
Paediatrics	0.83	0.66	0.74	741
Pain Management	0.90	0.55	0.68	147
Physiotherapy	0.80	0.83	0.82	1348
Rapid diagnostic centre	0	0	0	31
Rehabilitation	0.95	0.83	0.89	128
Rheumatology	0.90	0.83	0.86	430
Thoracic medicine	0.92	0.95	0.94	594
Urology	0.93	0.96	0.95	1766
Vascular surgery	0.81	0.65	0.72	146
Gynaecology	0.95	0.96	0.96	2024
Neurology	0.85	0.78	0.81	441
Geriatric medicine	0.92	0.54	0.68	81
Accuracy			0.88	22226
Macro average	0.85	0.71	0.76	22226
Weighted average	0.88	0.88	0.88	22226

A report using the non-linear RBF kernel (Table 42) produces results that lie between the linear SVM and the logistic regression models. The same trends can be seen where the classes with less than one hundred members in the testing set have a lower F1-Score than those with a greater number of examples. Using the rapid diagnostic centre class as the example again, the RBF scores one hundred percent on precision, meaning that every case that was attributed to that label was correct. However, the recall is only six percent which means that a majority of the cases that belonged to the rapid diagnostic centre class were actually classified by the model as a different label.

Table 42 Classification Report: RBF SVM on referral dataset

	precision	recall	f1-score	support
Cardiology	0.95	0.96	0.96	682
Care of the elderly	0.76	0.44	0.56	59
Clinical Immunology	0.91	0.76	0.83	88
Clinical neuro-physiology	0.88	0.50	0.64	44
Community Orthopaedic	0.78	0.32	0.45	393
Dermatology	0.95	0.97	0.96	3017
Dietetics	0.97	0.90	0.93	346
endocrinology	0.84	0.79	0.82	201
ENT	0.93	0.98	0.95	2551
Gastroenterology	0.79	0.85	0.82	987
General medicine	0.83	0.60	0.70	202
General Surgery	0.89	0.92	0.91	2875
Haematology (clinical)	0.93	0.77	0.84	146
Nephrology	0.95	0.72	0.82	87
Ophthalmology	0.91	0.75	0.83	126
Oral/Maxillo facial surgery	0.91	0.66	0.77	218
Orthopaedic	0.82	0.95	0.88	2327
Paediatrics	0.89	0.69	0.78	741
Pain Management	0.90	0.65	0.75	147
Physiotherapy	0.84	0.85	0.84	1348
Rapid diagnostic centre	1.00	0.06	0.12	31
Rehabilitation	0.98	0.84	0.90	128
Rheumatology	0.92	0.87	0.90	430
Thoracic medicine	0.94	0.96	0.95	594
Urology	0.95	0.97	0.96	1766
Vascular surgery	0.83	0.73	0.78	146
Gynaecology	0.96	0.97	0.96	2024
Neurology	0.88	0.82	0.85	441
Geriatric medicine	0.91	0.65	0.76	81
Accuracy			0.90	22226
Macro average	0.90	0.76	0.80	22226
Weighted average	0.90	0.90	0.90	22226

5.3.2.2 Classification using neural network approaches

As shown in Section 5.3.2.1, the word embedding approach struggles on this dataset when compared to the frequency-based approaches of TF-IDF and bag of words. Unfortunately, due to the size of the dataset's vocabulary it is impractical to transform the sparse matrices into usable inputs for neural network testing. Variations on linear neural network architectures were applied to this dataset with Doc2Vec embeddings as an input. With the information previously learnt from the urology dataset, dropout layers were included from the beginning to reduce the chance of overfitting.

Table 43 Linear Neural Network architectures used on referral dataset

Architecture	Validation Accuracy
Input layer: 128 nodes Hidden Layer: 64 nodes Output Layer: 29 nodes	.81
Input layer: 2048 nodes Hidden Layer(s): 1024 – 256 - 64 Output Layer: 29 nodes	.86
Input layer: 4096 nodes Hidden Layer: 2048 – 1024 – 256 – 64 Output Layer: 29 nodes	.78
Input layer: 256 nodes Hidden Layer(s): 512 - 512 Output Layer: 29 nodes	.84

Adapting the other neural network architectures to this dataset did not represent the data in a meaningful way. The previously discussed issue with the RNN was exemplified with the larger dataset; a single epoch taking six and a half hours to run. On the urology dataset, the RNN produced a comparable accuracy to the non-neural network models. It barely achieved an accuracy of .64 on the general referral letters dataset despite a five-day runtime. The CNN fared even worse than the RNN. Although the training was fast in comparison to the RNN, it would often fail to converge and models that did successfully begin separating the data would give accuracies between .40 and .60. Accuracies that are not high enough to be considered useful when traditional models are offering results above ninety percent.

The final neural network architecture used in this project the transformer architecture. The project employed RoBERTa (Liu, et al., 2019), a pre-trained model built upon the original BERT architecture (Devlin, et al., 2018) and has been used as the basis for domain specific medical language training in other publications (Monea & Marginean, 2021) including the Biomedical Language Understanding and Reasoning Benchmark (BLURB) (Gu, et al., 2021). Additionally, the BioLinkBERT (Yasunaga, et al., 2022) model which holds the highest position on the BLURB leaderboard as of June 2022 has been used to compare the result with the base RoBERTa model. The results of training these transformers on the referrals dataset are shown in Table 44.

Table 44 Results of applying transformer architectures to the referral dataset

Epochs	RoBERTa-base		BioLinkBERT-base	
	MCC	F1-Score	MCC	F1-Score
1st	0.833	0.835	0.883	0.891
20	0.924	0.930	0.925	0.931

The transformer architecture outperformed every other neural network based approach by a significant margin. The final F1-Scores are higher than all other machine learning techniques tested in this thesis for both the specialist and non-specialist trained models. The only issue that arises during the training of these models using existing libraries is the storage space required to save the model at different points in the training process. Each checkpoint required 1.39GB of storage and, with a large dataset like the referral dataset, there were six checkpoints for every epoch.

5.3.2.3 Impact of feature reduction techniques when classifying

The vocabulary size of the referral dataset is incredibly large due to the number of cases and variety of different specialities present. The number of features causes many of the clustering and classification tasks to use a significant amount of processing time and power. With some models, like the convolutional neural network, it removed the ability to classify the system entirely.

The same approach to the one set out in Section 5.2.2.3 was taken, wherein a modified version of feature elimination was carried out via thresholding instead of recursion. The absolute weights for all features were taken from a linear support vector machine, and at each stage the features that fell below the threshold were removed. The resulting changes to the vocabulary size are detailed in Table 45. The accuracy of a model implementing SVD reduction has not been implemented on this dataset due to memory limitations.

Table 45 Referral vocabulary size and resultant F1-Scores when thresholding

Threshold Weight	Vocabulary Size	F1-Score (COEF)	F1-Score (TF-IDF)
0.0	309325	0.91609	0.91609
0.1	266991	0.91613	0.91644
0.2	232686	0.91634	0.91615
0.3	199571	0.91607	0.91569
0.4	166804	0.91552	0.91575
0.5	137137	0.91494	0.91540
0.6	111842	0.91542	0.91493
0.7	91172	0.91500	0.91476
0.8	74518	0.91374	0.91294
0.9	61313	0.91275	0.91122
1.0	50805	0.91067	0.91129
1.1	42605	0.90961	0.90961
1.2	35973	0.90803	0.90821
1.3	30888	0.90696	0.90692
1.4	26724	0.90496	0.90603
1.5	23266	0.90260	0.90355
1.6	20447	0.90102	0.90377
1.7	18113	0.89986	0.90237
1.8	16121	0.89842	0.90016
1.9	14435	0.89743	0.89979
2.0	12990	0.89475	0.89850
2.1	11709	0.89220	0.89617
2.2	10626	0.89078	0.89445
2.3	9718	0.88971	0.89302
2.4	8932	0.88772	0.89133
2.5	8243	0.88606	0.89000
2.6	7611	0.88637	0.88850
2.7	7006	0.88493	0.88660
2.8	6514	0.88340	0.88570
2.9	6016	0.88208	0.88297
3.0	5570	0.88009	0.88101
3.1	5185	0.87871	0.88033

3.2	4844	0.87722	0.87684
3.3	4524	0.87688	0.87483
3.4	4235	0.87451	0.87374
3.5	3957	0.87287	0.87218

The results show that if the vocabulary were to be reduced to the same threshold as the urology dataset (1.0), then the vocabulary size would be reduced by 73.5 percent for a reduction in accuracy by only 0.00542 percent. However, reducing the vocabulary size also reduces the accuracy of models because there are some noisy features. Using a threshold of 0.1 or 0.2 for features provides a better F1-Score with both TF-IDF max features and linear coefficients.

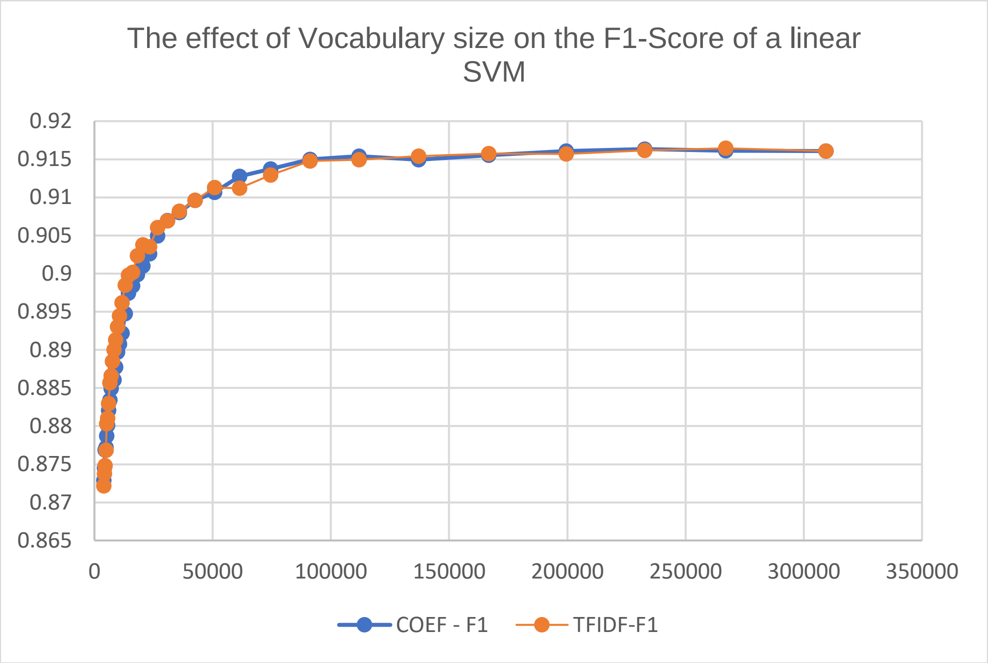


Figure 5.22 F1-Score changes using SVM coefficients and TF-IDF max features

Figure 5.22 graphs the information from Table 45. The line follows the same trend that was shown in the urology dataset. The model's accuracy fluctuates around the same point within one or two percent until reducing the vocabulary to a small number of features. It reaches this point at an earlier threshold than previously seen; in the referral dataset it occurs at 0.8 in comparison to 1.6 in the urology dataset. This is due to the number of classes evaluated for this task and the size of the initial vocabulary. Words that are still useful for classification but appear across multiple documents will have a lower initial weight value to the TF-IDF method of vectorisation. For the weights assigned by the SVM itself, the impact of a feature depends on how much it effects the classification towards a single class. Features that exist across multiple classes will not convey the same level of positive or negative weighting as was seen in the urology dataset. However, for certain classes like the rapid diagnostic centre or for separating community orthopaedic from normal orthopaedic, these overlapping features are the most important.

5.3.2.3.1 Effects of stemming and lemmatisation on general referral letters

As previously discussed in this thesis, the use of stemming and lemmatisation has been avoided due to the specific language used within medical texts including treatments, symptoms, and tests. This reasoning for not adjusting the original texts has also been seen in Section 4.3□o, where the clinical named entity recognition mistook several specific features as other medical terms.

Table 46 The effects of pre-processing on model performance

	precision	recall	f1-score
Baseline	5	3	5
Stemmed	6	5	3
Lemmatised	3	3	4

Table 46 shows which method of presenting the data resulted in the greatest accuracy for each performance metric. Focussing on the F1-Score, which is the metric used to evaluate the models in this thesis, only twelve out of the twenty-nine categories have an F1-Score that is higher than the other two methods. The full classification reports for the model's using stemming and lemmatisation can be found in Appendix 5 Classification reports for stemming and lemmatising referrals.

5.3.2.4 Classification techniques applied around prioritising patients

Patient prioritisation is another element of the dataset that is valued by the company partner on this project. As stated in Section 4.2.3.3 with Figure 4.7, there is a significant difference in the priority a general practitioner will initially assign to a patient and the resultant priority assigned by the consultant. This discrepancy has shown to change alongside the version numbers attached to each of the cases, with the most dramatic change appearing between the initial letter (version number 1) and the first iteration (version number 2). The support vector machine that produced the best result on the speciality classification problem has been used for the prioritisation problem as well. Table 47 shows the results of modelling with all the notes and a subset of the dataset that only looks at the initial referral letters.

Table 47 Performance prioritising letters for all notes and just initial notes

	All notes			Version 1 only		
	precision	recall	f1-score	precision	recall	f1-score
Routine	0.87	0.97	0.92	0.88	0.96	0.92
Urgent	0.87	0.55	0.68	0.87	0.56	0.68
USC	0.84	0.88	0.86	0.84	0.89	0.86
Accuracy	0.87			0.87		

Across the entire dataset, the GP priorities align with the consultant priorities ninety one percent of the time. The support vector machine does not manage to achieve the same accuracy as a medical professional assigning priorities to a patient. However, due to the method of communication between the GP and consultant, any case that has a version number of two or greater is more likely to match as it is a combination of both clinician's opinions. To that extent the most important cases within the dataset to look at are the initial letters (labelled as version one). The support vector machine approach matches the GP's accuracy for cases associated with routine and USC cases but struggles on urgent cases. This mirrors the issues shown in Figure 4.7, the most common change in case prioritisation between GP and consultant occurring within the urgent class.

While an accuracy of eighty seven percent is acceptable as a research goal, an increase in accuracy is needed to make the model viable for implementation as a medical system. With the goal of providing a system that can help clinicians rather than replace clinicians, a way to tackle this issue is to reduce the number of letters classified to only those cases where the system is confident in the prediction.

Table 48 Accuracy of the SVM at different confidence levels

Confidence	Accuracy	Priority unchanged	Priority upgraded	Priority downgraded	Total prioritised	Percentage of letters requiring clinical review
> 60%	88%	68.41%	2.34%	7.25%	78%	22%
> 70%	91%	59.99%	1.25%	4.75%	66%	34%
> 80%	94%	47.89%	1.20%	2.50%	51%	49%
> 90%	96%	25.68%	1.46%	0.87%	28%	72%
> 95%	98%	10.44%	0.35%	0.21%	11%	89%
> 98%	98%	1.92%	0.06%	0.03%	2%	98%

Table 48 indicates how the confidence level effects the accuracy of the model. Depending on the threshold chosen, the impact the system has as a support tool changes. Choosing a high confidence threshold like ninety five percent results in an accuracy value of ninety eight percent. The number of prioritised letters decreases alongside this increase in accuracy, with eleven percent of the documents classified by the system. Given the dataset analysed, an 11% reduction is 12,224 notes. This represents a significant reduction in work otherwise invested by clinicians.

5.3.3 Observations

Experiments carried out on the referral dataset proved the hypothesis set out at the end of 5.2.3, that the main issue with classification during the testing on the urology dataset was due to the sub-specialities existing within close proximity of each other. A wider set of features present in the referral dataset showed that multiple different methods of classification could be used to successfully represent the relationships between features and an associated speciality.

Again, feature reduction techniques were able to reduce the size of the dataset without impacting the performance of the models until a certain point. Using the thresholds calculated from the linear SVM's coefficients, a reduction of thirty five percent of the vocabulary could be achieved before the model suffered a loss in accuracy. The impact of stemming and lemmatisation was minimal on the data, and not worth implementing in a full system. Doing so would incur a computational overhead to the system wherein every new input by a clinician would require an added level of pre-processing.

The results on the prioritisation classes are useful for presenting the idea that a clinician need not review every document for that purpose. While the model is not fit for purpose when looking at every inputted document, incorporating a high confidence threshold results in a model that could reduce the workload of clinicians by upwards of eleven percent.

5.4 Creation of a multiple output explainable system

Research question one is to create a generalised repository of knowledge that could serve as a basis for a clinical support system, rather than a classification model that replaces the medical professional. Therefore, the scenario presents a unique opportunity. A system can be created that both displays multiple potential outputs to the clinician interacting with the system and gives reasoning behind which features have influenced that decision.

To do so, however, eliminates the prospect of using deep learning models as the methodologies are black box. No information can be gleaned about the actual computations that are carried out between the input vectors and the output classification. Weights can be extracted, but they will not directly correlate to any feature in the vocabulary. Instead, the focus will be on the white box techniques that have shown to perform as well as these deep learning techniques: support vector machines and logistic regression. Figure 5.23 contains the code which extends a pipeline to output the top three classes found on a testing set to a pandas dataframe (top_class).

```
#Calculate probabilities from the model

proba = model.predict_proba(X_test)
best_probs = np.argsort(-proba, axis=1)[:, :3]


#Create a dataframe with the probabilities for each class
top_class = model.classes_[best_probs]
top_class_df = pd.DataFrame(data=top_class)
results = pd.DataFrame(y_test, columns=['correct'])
results = pd.merge(results, top_class_df, left_index=True, right_index=True)
results['indices'] = indices_test

# Fetch top 3 results
top3 = [
    (results.iloc[:, 0] == results[0]),
    (results.iloc[:, 0] == results[1]),
    (results.iloc[:, 0] == results[2])]
top3_choices = [1, 1, 1]
# Fetch top 2 results
top2 = [
    (results.iloc[:, 0] == results[0]),
    (results.iloc[:, 0] == results[1])]
top2_choices = [1, 1]
# Fetch Top result
top1_conditions = [(results.iloc[:, 0] == results[0])]
top1_choices = [1]

# Create the success columns
results['Top 3 Successes'] = np.select(top3, top3_choices, default=0)
results['Top 2 Successes'] = np.select(top2, top2_choices, default=0)
results['Top 1 Successes'] = np.select(top1_conditions, top1_choices, default=0)


# Output the accuracy of the system when looking at the top x results
print("Predicted Accuracy using .predict_proba(top 1)= ", sum(results['Top 1 Successes'])/results.shape[0])
print("Predicted Accuracy using .predict_proba(top 2)= ", sum(results['Top 2 Sucessses'])/results.shape[0])
print("Predicted Accuracy using .predict_proba(top 3)= ", sum(results['Top 3 Successes'])/results.shape[0])
```

Figure 5.23 Code used to output top probabilities to a user

This dataframe is then merged with the actual output (*y_test)* and the original document number (indices_test) to create the dataframe shown in Table 49.

Table 49 Results dataframe after initial merging

Correct	0	1	2	Indices
Actual class label	Predicted label	Second highest predicted label	Third highest predicted label	Document number

From this dataframe, three lists are made: top3, top2 and top1. Each list creates a binary output as to whether the class number predicted in a column matches the correct column. These lists alongside the choice lists (top3/top2/top1_choices) can be used with numPy's select function to return a binary array representing whether the correct class was successfully found in the top *n* predictions by the model. In the example these arrays have been assigned to columns in the results dataframe and the effect of the conditions shown in Table 50.

Table 50 Success Columns added to Results dataframe

Top 3 Successes	Top 2 Successes	Top 1 Successes
1: if prediction in top 3 0: if prediction not in top 3	1: if prediction in top 2 0: if prediction not in top 2	1: if prediction in top 1 0: if prediction not in top 1

To achieve this result requires models that can output interpretable probabilities. If using a library like scikit-learn (Pedregosa, et al., 2011) the default linearSVC (SVM) model cannot output probabilities. Instead, the SVC model must be employed with a linear kernel and the probability parameter set to true. This method enables the use of Platt scaling (Platt, 1999) to transform the outputs of the support vector machine to a probability distribution. The outputs of the accuracy calculations with these two models with a different number of predictions is shown in Table 51.

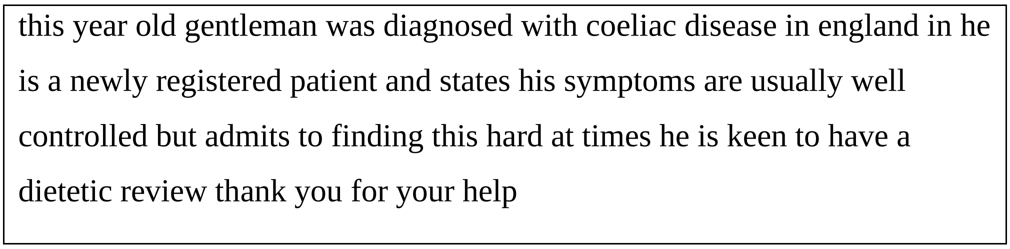
Table 51 Testing if the correct answer is within the top three probabilities using SVM and Logistic regression

Number of predictions given	SVM	Logistic Regression
One	0.9069	0.8829
Two	0.9763	0.9649
Three	0.9919	0.9844

The results in Table 51 show that when classifying against twenty-nine categories of medical speciality, the support vector machine is ninety-nine percent effective when three possibilities are outputted to the user. As expected, a percentage of the misclassifications are due to overlapping medical terms between classes amongst other problems. The additional benefit to approaching the problem in this way besides the accuracy score is that the probabilities given to each output are visible to the end user. This allows the system proposed in Section 5.4.2 to output the predicted values to the screen to support a decision made by a clinician.

5.4.1 Problems within letters resolved by offering multiple predictions

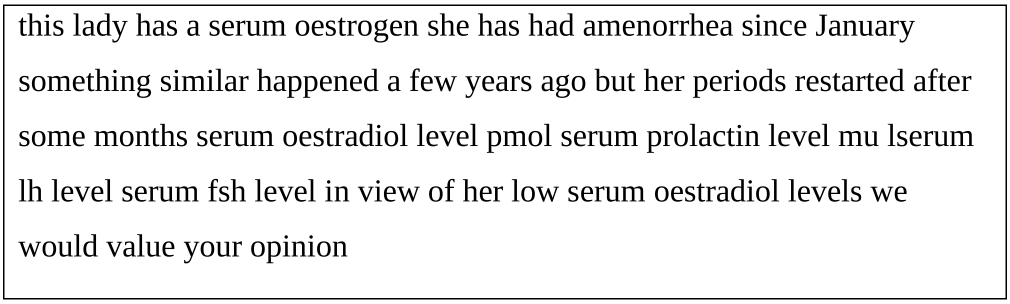
This section outlines examples of misclassifications in the original classification system that have been resolved by offering an alternative option. It outlines some of the issues the models encountered due to the medical nature of the dataset.



this year old gentleman was diagnosed with coeliac disease in england in he is a newly registered patient and states his symptoms are usually well controlled but admits to finding this hard at times he is keen to have a dietetic review thank you for your help

Figure 5.24 Coeliac disease letter

The first example, Figure 5.24, was assigned as a gastroenterology letter instead of a dietetics letter. This example highlights an issue a computer has when interpreting the intention of a letter. The condition *coeliac disease* is a gastroenterology problem, but the intended result is to see a dietician to manage the condition. By offering an alternative the system was able to offer both gastroenterology and dietetics as speciality choices.



this lady has a serum oestrogen she has had amenorrhea since January something similar happened a few years ago but her periods restarted after some months serum oestradiol level pmol serum prolactin level mu lserum lh level serum fsh level in view of her low serum oestradiol levels we would value your opinion

Figure 5.25 Amenorrhea letter

Figure 5.25 shows a case that potentially had the wrong speciality assigned by the GP. In the dataset, the speciality assigned was General medicine. A speciality that had been discussed in the presenting complaints dataset (see Section 4.2.1) as a universal label when a case does not fit a more specialist class. Using information found via the NHS website (National Health Service, 2019), a person suffering from amenorrhea should be referred to a gynaecologist or an endocrinologist. Both of these options were presented as the number one and number two predictions, with general medicine being the third prediction.

5.4.2 Potential for producing an integral system

All clinicians within Wales interact with the Wales Clinical Portal (WCP) for a variety of reasons including dealing with referrals. For a machine learning system like the one discussed in this thesis to be implemented into the NHS as a tool, it must be integrable into this portal. Figure 5.26 presents a rough outline of how such a tool could fit into the referral section of the clinical portal. Once information has been entered into the Presenting complaint section of the referral, a speciality advice button could provide options for assigning a specific referral speciality when hovered over. These options will then highlight key terms present within the text that indicate that the case should be referred to a consultant belonging to that speciality.

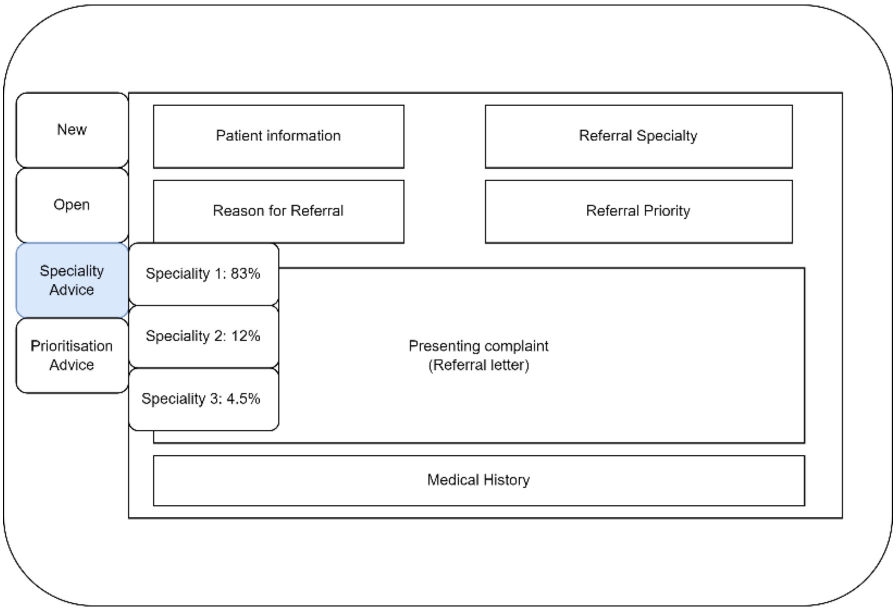
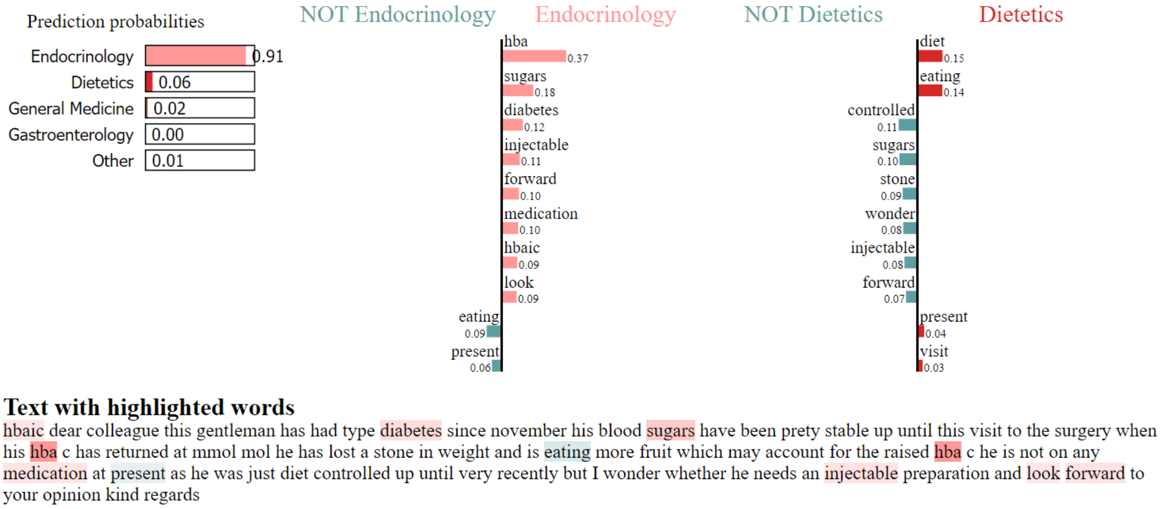


Figure 5.26 Prototype of integrated support tool

As per research question three, any support tool that would be incorporated into the Welsh clinical portal would need to be computationally inexpensive and intuitive. A support tool that hinders the speed of a clinician to perform their job would be impractical and ignored.

5.4.2.1 How existing machine learning explainers provide classification insights

An indication into how such a system would function can be garnered from the use of existing machine learning explainers. Both LIME (Ribeiro, et al., 2016)) and SHapley Additive exPlanations (SHAP) (Lundberg & Lee, 2017) offer approaches to model interpretation. Out of the two approaches, LIME presents the information surrounding a single prediction in a more understandable format. Figure 5.27 shows an example of how LIME interprets the case passed into an explainer, and outputs weights associated with individual features.



Text with highlighted words

hbaic dear colleague this gentleman has had type diabetes since november his blood sugars have been prety stable up until this visit to the surgery when his hba c has returned at mmol mol he has lost a stone in weight and is eating more fruit which may account for the raised hba c he is not on any medication at present as he was just diet controlled up until very recently but I wonder whether he needs an injectable preparation and look forward to your opinion kind regards

Figure 5.27 LIME showing features contributing to an endocrinology classification

While LIME produces an output in a readable format, it is important to acknowledge that the associated probabilities, features, and weights are produced by creating a simplified version of the original dataset. Any distance calculations performed will reflect this new dataset and the linear model associated with LIME, rather than the original predictions modelled. As such, the outputs may not represent the information correctly if the original model is using a more complex classifier.

5.5 Chapter conclusion

The results outlined in this chapter communicate the importance of having high quality data and enough variation between the class labels. Starting with the Accident and Emergency presenting complaints dataset, the modelling results improved with each subsequent dataset. The chapter shows the importance of each choice made across the entire modelling process, including the impact each choice has on both accuracy values and computing power requirements.

The results themselves reflect the ability to classify data effectively against medical specialities and patient prioritisations with the general referral letters dataset. The methods chosen in this research achieve results of 92% and 87% against the two sets of class labels. With regards to the patient specialities, diverging from the common path of classification and instead producing a probabilistic output of two or three options results in an accuracy of 99%. For patient prioritisation, as there are only three labels to be assigned, the research instead looks at employing confidence thresholds. By doing so the research undertaken shows that 11% of letters could be automatically prioritised by a system with 98% accuracy.

This chapter also outlines an alternative to Recursive Feature Elimination (RFE), using absolute values of SVM weight vectors to rank features within a vocabulary. The resultant output has shown a comparative feature reduction ability to other methods and can be performed on large datasets that would otherwise require segmentation to use PCA.

The next chapter concludes this thesis, highlights contributions to knowledge, offers suggestions for future work and presents final comments.

Chapter 6 Conclusion

This concluding chapter presents a summary and assessment of the research accomplished in this thesis. It draws on the research aims and objectives that were established in the introductory chapter of this thesis alongside the results shown in individual chapters to justify the conclusions.

6.1 Research objectives

The main aim of this research was to demonstrate the effectiveness of document classification within the clinical domain without the need of external clinical dictionaries. The research in this thesis has successfully shown that words and phrases contained within a medical letter between a general practitioner and a consultant provides enough information to model relationships about the associated medical speciality and patient priority. The success of this aim was attributed to the completion of four research objectives.

- To investigate machine learning and its existing use within the clinical domain alongside the fundamental decision making of clinicians when referring patients to hospital.

This objective was completed across this thesis. Beginning in Chapter 2, this thesis outlined the process that a patient undergoes to receive a referral from a general practitioner to a consultant. This includes indicators (symptoms, test results, etc.) used by clinicians to determine the specialist required and urgency of the case. It then goes onto introduce natural language processing and the classification pipeline being used for this research.

The initial investigation into Natural Language Processing within the clinical domain found that the most prominent subtask was named entity recognition. This is due to the nature of medical texts which tend to include abbreviations for items like pharmaceuticals and test results. Publications for clinical document classification were found to mostly include a pre-trained NER system during feature generation, with only a few newer publications using systems that did not include medical ontologies (see Section 2.4). The applicability of these pre-trained NER systems was tested in Section 4.2. The outcomes of which found that issues can occur when implementing such a system. On the examples tested, the annotations were often obscure and unrelated to the rest of the document.

- To assess the impact that different feature generation techniques can have on the ability for a machine learning algorithm to model relationships in the data.

Chapter 3 introduced the two methods of feature generation: frequency-based models and context-based models. Within these sections, the main techniques used for feature generation were explained including the processes for transforming raw text data into numbers that can be understood by a computer.

Chapter 3 also compares both approaches, explaining the benefits and issues that can occur when using them. This comparison was taken further in Chapter 5 wherein the classification results for both the urology (see Section 5.2.2) and general referral dataset (see Section 5.3.2) included accuracy comparisons using different feature generation methods like TF-IDF and Doc2Vec.

- To develop a natural language processing classification pipeline that takes in raw text data and outputs labels according to medical speciality or priority.

Each stage of a classification pipeline has been expressed in this thesis. For the explicit development of a pipeline, Section 4.2 and Section 4.3 demonstrate the initial pre-processing of each dataset, alongside the models chosen for classification and their associated hyperparameters. Chapter 5 then shows the results of the created pipeline for both patient priority and medical speciality. Following this, Section 5.4 describes the creation of a multiple output system that explains the decisions made by a classifier to an end user. It shows the benefits of adding choices to the final decision in terms of both accuracy improvements as well as highlighting problems that may arise within clinical text (See Section 5.4.1).

- To evaluate the contributing factors to a successful classification including feature generation, model selection and feature reduction.

The contributing factors of classification are displayed in Chapter 5 for each of the datasets used in testing. The results firstly compare the different feature generation algorithms and their impact on accuracy for the different machine learning models.

Following this, an evaluation of traditional machine learning techniques compared to neural networks is carried out. The ability of each model to classify against the pre-labelled data is tested and considerations are made in relation to whether the benefits of being able to understand exactly how a model has predicted an outcome outweighs a drop in overall accuracy.

Finally, Section 5.2.2.3 and Section 5.3.2.3 evaluate different methods of feature reduction to reduce the time and power requirements needed to train a large machine learning model whilst not affecting the performance of a model. The sections discuss existing methods like PCA and thresholding during feature generation as well as introducing a method of feature reduction using weights produced by a linear SVM.

6.2 Contributions to Knowledge

The research described in this thesis falls under the categories of medical subdomain classification with an onus on presenting the information in a format that a clinical professional interested in the process can follow. Contributions made in regard to this topic are stated in this section.

6.2.1 Flexibility in research outputs

This research shows the potential for machine learning models to support existing manual processes within the clinical domain. It incorporates multiple different machine learning techniques to fulfil the goal of classifying medical data, achieving this goal accurately with both unsupervised and supervised learning. It also contains comparisons between each of the approaches and the benefits/downsides to using them.

The research also shows the advantage of not being restricted to a single classification. By eliminating the requirement for the system to only show its most confident classification choice, the accuracy of the system was increased significantly (to 99%). The only remaining letters that the system then struggled to classify had underlying issues and are outlined in Section 5.4.1.

6.2.2 Interpretation of a practical dataset

Natural language processing tasks in the clinical domain often struggle to find an appropriate dataset for the hypothesis being tested. This research uses three datasets taken from within the DHCW data warehouse, which consist of live patient data, to ensure that the features used for training mirror what would be found in an actual referral letter. Furthermore, the key difference that sets this research apart from other publications on clinical document classification is that the selection of input data was random. No modifications were made to the data to better accommodate classification, such as limiting the number of classes or manually curating the data to fit a specific goal. The only condition enforced on the data collection process was a date range of two years (2017-2019). As previously discussed in Section 2.4, this method of data procurement is objectively more representative of an actual healthcare system than some existing publications.

6.3 Future Work

This section describes the next steps that could be taken to further this research, based on the results shown in this thesis.

6.3.1 Implementation of a system into the Welsh clinical portal

With the main research aim achieved, the next stage would be to determine the transferability of the research into a consumer environment. While this thesis presents sufficient results to show the models worth in a research environment (see Section 5.4), a public implementation would be required to take it a step further. To do so, however, will require the system to be created using a user-centric design, with a substantial number of hours spent shadowing and liaising with clinicians on both ends of the referral process to understand the information needed to ensure patient safety. The concerns around patient safety are explained in The Digital Doctor (Wachter, 2015). However, some of the worries surrounding the replacement of clinicians are already covered by the support aspect of the proposed system.

The other need for clinical input on a consumer facing system questions whether the full picture of a person’s case can be extracted by the presenting complaint alone. With the integration of computing systems into healthcare, what was once the entire referral letter has been broken up into several different text boxes that are filled in by the general practitioner or consultant separately. There may not be a need to include the information about the patient’s medical history in the presenting complaint. By doing so a consultant may find indicators that a patient needs to be assigned a different speciality or prioritisation in the medical history that was overlooked by a general practitioner. The presence of certain symptoms directing a system towards a specific speciality may exist because of such an indicator. As an example, blood in urine may be caused by a urological issue but can also be a result of a history of smoking. Again, this level of knowledge would need to be achieved by shadowing and liaising with clinicians directly and appropriately.

Existing classification systems within a clinical context do exist, primarily image classification systems that aim to detect the existence of cancer indicators from photos and radiology reports. None of these systems replace the clinicians or technicians working with the original images. Human input is required before and after the classification model decides what it believes to be a melanoma or tumour. It is this prospect that leads to the belief that the same type of system can be influential with textual documents to improve the speed of clinical pathways undertaken by patients.

6.4 Final Remarks

The natural language processing pipeline and its associated techniques demonstrated in this research demonstrates an effective methodology for approaching a machine learning task with an unstructured clinical dataset. The research shows that there are fundamental relationships present within the clinical referral letters evaluated that can be used to interpret key labels such as speciality department and patient prioritisation. The accurate results achieved across a variety of machine learning techniques like support vector machines and artificial neural networks show the strength of these relationships, avoiding the use of external dictionaries that are commonly found within literature in this area.

The research also establishes an alternate approach to feature elimination using the absolute values of the SVM weight vectors as thresholding points. This is in contrast to existing approaches that approach the feature elimination process recursively. Such techniques can then be combined to find the best vocabulary size between the two most accurate thresholds. The results of thresholding the data using SVM weight vectors show comparable results with other feature reduction techniques like TF-IDF values and PCA. Unlike PCA there is no data transformation that creates a large memory overhead, meaning that the method shown in this research can be used on large datasets under strict hardware constraints.

Due to the highly technical language used within clinical data representing symptoms, medical procedures, and medication, creating a system that can perfectly determine the nature of a patient’s problem is incredibly difficult. However, the approach taken in this research shows that a system that offers one or two alternatives to the best match presents the user with choices that represent over 99% accuracy in terms of matching case description to outcome. The system’s ability to capture and share medical expertise makes available a wealth of useful information to help in decision making. Such a system has the potential to better direct more patients to the right specialist earlier in the process than is currently the case. In addition, clinical expertise captured in this way may be particularly impactful in developing countries where the availability of second opinions may be sparse.

References

Abadi, M. et al., 2015. *TensorFlow: Large-scale machine learning on heterogeneous systems.* [Online]
Available at: Software available from tensorflow.org

Abdi, H. & Williams, L. J., 2010. Principal component analysis. *WIREs Computational Statistics,* 2(4), pp. 433-459.

Aho, A. V., 1991. Algorithms for finding patterns in strings, Handbook of theoretical computer science (vol. A). In: *Algorithms and Complexity.* Cambridge, MA.: MIT Press.

Aizawa, A., 2003. An information-thoeretic perspective of tf-idf measures. *Information Processing & Management,* pp. 45-65.

Alexandrescu, A. & Kirchhoff, K., 2006. *Factored Neural Language Models. In Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers.* New York City, USA, Association for Computational Linguistics.

Aronson, A. R., 2001. *Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program..* s.l., s.n., pp. 17-21.

Bagga, A. & Baldwin, B., 1998. Entit-based cross-document coreferencing using the Vector Space Model. *s.l., s.n..*

Baldi, P. et al., 2000. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics,* 16(5), pp. 412-424.

Baroni, M., Dinu, G. & Krusewski, G., 2014. *Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors.* East Stroudsburg PA, ACL, pp. 238-247.

Beam, A. L. et al., 2020. Clinical Concept Embeddings Learned from Massive Sources of Multimodal Medical Data. *Biocomputing,* pp. 295-306.

Behera, B., Kumaravelan, G. & Prem Kumar B., 2019. Performance Evaluation of Deep Learning Algorithms in Biomedical Document Classification. *2019 11th International Conference on Advanced Computing (ICoAC),* pp. 220-224.

Bengfort, B. & Bilbro, R., 2019. Yellowbrick: Visualising the Scikit-Learn Model Selection Process. *The Journal of Open Source Softwarew,* 4(35).

Bengio, Y., Ducharme, R., Vincent, P. & Jauvin, C., 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research 3,* pp. 1137-1155.

Bengio, Y., Frasconi, P. & Simard, P., 1993. *The problem of learning long-term dependencies in recurrent networks.* San Francisco, IEEE Press.

Bengio, Y., Simard, P. & Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks,* 5(2), pp. 157-166.

Bird, S., Klein, E. & Loper, E., 2009. *Natural language processing with Python: analyzing text witht hte natural language toolkit.* s.l.:O'Reilly Media, Inc..

Blum, A., Hopcroft, J. & Kannan, R., 2013. *Foundations of Data Science.* s.l.:Cambridge University Press.

Bodenreider, O., 2004. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Research,* 32(1), pp. D267-D270.

Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T., 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics,* Volume 5, pp. 135-146.

Boole, G., 1854. *An investigatiopn of the laws of thought: on which are founded the mathematical theories of logic and probabilities.* London: Cambridge: Macmillan and Co..

Boser, B. E., Guyon, I. M. & Vapnik, V. N., 1992. *A training algorithm for optimal margin classifiers.* New York, COLT '92: Proceedings of the fifth annual workshop on Computational learning theory.

Bottou, L., 2012. Stochastic gradient descent tricks. In: *Neural networks: Tricks of the trade.* Berlin: Springer, pp. 421-436.

Bradley, P. S. & Fayyad, U. M., 1998. *Refining initial points for k-means clustering.* s.l., ICML.

Buchan, K., Filannino, M. & Uzuner, Ö., 2017. Automatic prediction of coronary artery disease from clinical narratives. *Journal of Biomedical Informatics,* pp. 23-32.

Byrd, R. H., Lu, P. & Zhu, C., 1995. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing,* 16(5).

Caliński, T. & Harabasz, J., 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1), pp. 1-27.

Cao, F., Liang, J. & Jiang, G., 2009. An initialization method for the K-Means algorithm using neighborhood model. *Computers & Mathematics with Applications,* 58(3), pp. 474-483.

Cao, L. J. et al., 2003. A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine. *Neurocomputing,* 55(1-2), pp. 321-336.

Castro, S. M. et al., 2017. Automated annotation and classification of BI-RADS assessment from radiology reports. *Journal of Biomedical Informatics,* Volume 69, pp. 177-187.

Cho, K., van Merrienboer, B., Bahdanau, D. & Bengio, Y., 2014. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv.*

Chollet, F. &. O., 2015. *Keras.* [Online]
Available at: https://keras.io

Clark, C. et al., 2007. Identifying smokers with a medical extraction system.. *Journal of the American Medical Informatics Association : JAMIA*, 15(1), pp. 36-39.

Cocos, A., Qian, T., Callison-Burch, C. & Masino, A. J., 2017. Crowd control: Effectively utilizing unscreened crowd workers for biomedical data annotation. *Journal of Biomedical Informatics,* Volume 69, pp. 86-92.

Cohen, K. B. et al., 2016. Methodological Issues in Predicting Pediatric Epilepsy Surgery Candidates through Natural Language Processing and Machine Learning. *Biomed Inform Insights,* Volume 8, pp. 11-18.

Dale, R., 2016. The return of the chatbots. *Natural Language Engineering,* 22(5), pp. 811-817.

Dale, R., Moisel, H. & Somers, H., 2000. *Handbook of Natural Language Processing.* Manchester, England: Marcell Dekker Inc..

David, A. & Sergei, V., 2007. *K-means++: the advantages of careful seeding.* s.l., In Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms.

Debie, E. & Shafi, K., 2019. Implications of the curse of dimensionality for supervised learning classifier systems: theoretical and empirical analyses. *Pattern Analysis & Applications,* 22(2).

Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K., 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv.*

Dong-Harris, K., Patterson, O., Igo, S. & Hurdle, J., 2013. *Document sublanguage clustering to detect medical specialty in cross-institutional clinical texts. In DTMBIO '13: Proceedings of the 7th international workshop on Data and text mining in biomedical informatics.* s.l., s.n.

Duchi, J., Hazan, E. & Singer, Y., 2011. Adaptive Subgradient MEthods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research,* Volume 12, pp. 2121-2159.

Dumais, S., Platt, J., Heckerman, D. & Sahami, M., 1998. *Inductive learning algorithms and representations for text categorization.* New York, s.n.

Fan, Y. & Zhang, R., 2018. Using natural language processing methods to classify use status of dietary supplements in clinical notes. *BMC Medical Informatics and Decision Making volume.*

Faris, H. et al., 2020. Medical speciality classification system based on binary particle swarms and ensemble of one vs. rest support vector machines. *Journal of Biomedical Informatics.*

Firth, J. R., 1957. A Synopsis of Linguistic Theory 1930-55.

Fodeh, S. J. et al., 2018. Classifying clinical notes with pain assessment using machine learning. *Medical & Biological Engineering & Computing,* 56(7), pp. 1285-1292.

Foot, C., Naylor, C. & Imison, C., 2010. *The quality of GP Diagnosis and Referral,* s.l.: The King’s Fund.

Frades, I. & Matthiesen, R., 2010. Overview on Techniques in Cluster Analysis. In: *Bioinformatics Methods in Clinical Research. Methods in Molecular Biology (Methods and Protocols).* s.l.:Humana Press.

Furey, T. S. et al., 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics,* 16(10), pp. 906-914.

Fu, S. et al., 2020. Clinical concept extraction: A methodology review. *Journal of Biomedical Informatics,* Volume 109.

Gatt, A. & Krahmer, E., 2018. Survey of the state of the art in natural language generation: core tasks, applications and evaluation. *Journal of Artificial Intelligence Research,* 61(1), pp. 65-170.

Gehrmann, S. et al., 2018. Comparing deep learning and concept extraction based methods for patient phenotyping from clinical narratives. *PLoS ONE* , 13(2).

Glorot, X., Bordes, A. & Bengio, Y., 2011. *Deep Sparse Rectifier Neural Networks.* s.l., Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics.

Gorodkin, J., 2004. Comparing two K-category assignments by a K-category correlation coefficient. *Computational Biology and Chemistry,* 28(5-6), pp. 367-374.

Grandini, M., Bagli, E. & Visani, G., 2020. Metrics for Multi-Class Classification: an Overview. *arXiv.*

Grave, E. et al., 2018. *Learning Word Vectors for 157 Languages. In Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018).* s.l., s.n.

Gudivada, V. N. & Rao, C. R., 2018. *Computational analysis and understanding of natural languages.* Amsterdam: Elsevier.

Gustafson, E. et al., 2017. A Machine Learning Algorithm for Identifying Atopic Dermatitis in Adults from Electronic Health Records. *IEEE Int Conf Healthc Inform.,* pp. 83-90.

Guyon, I. & Elisseeff, A., 2003. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research* , pp. 1157-1182.

Guyon, I., Weston, J., Barnhill, S. & Vapnik, V., 2002. Gene Selection for Cancer Classification using Support Vector Machines. *Machine Learning,* Volume 46, pp. 389-422.

Gu, Y. et al., 2021. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. *ACM Transactions on Computing for Healthcare,* 3(1), pp. 1-23.

Harris, C. R. et al., 2020. Array Programming with NumPy. *Nature, 585,* pp. 357-362.

Harris, Z., 1954. Distributional Structure. *Word.*

Hassanpour, S., Bay, G. & Langlotz, C. P., 2017. Characterization of Change and Significance for Clinical Findings in Radiology Reports Through Natural Language Processing. *Journal of Digital Imaging,* pp. 314-322.

He, J. et al., 2004. *Initialization of cluster refinement algorithms: a review and comparative study.* s.l., IEEE International Joint Conference on Neural Networks.

Hinton, G., 2012. *Neural Networks for Machine Leanring: Lecture 6a,* s.l.: Coursera.

Hochreiter, S. & Schidhuber, J., 1997. Long Short-term Memory. *Neural Computation,* 9(8), pp. 1735-1780.

Honnibal, M. & Montani, I., 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing..

Hughes, M., Li, I., Kotoulas, S. & Suzumura, T., 2017. Medical text classification using convolutional neural networks. *Stud Health Technol Inform.,* pp. 46-50.

Joachims, T., 1998. *Text Categorization with Support Vector.* Berlin,) Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science, vol 1398. Springer.

Johnson, A. E. et al., 2016. MIMIC-III, a freely accessible critical care database. *Scientific Data.*

Joulin, A., Grave, E., Bojanowski, P. & Mikolov, T., 2017. *Bag of Tricks for Efficient Text Classification. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers.* s.l., Association for Computational Linguistics.

Karlsson, I. & Boström, H., 2014. *Handling Sparsity with Random Forests When Predicting Adverse Drug Events from Electronic Health Records. In IEEE International Conference on Healthcare Informatics (ICHI).* s.l., IEEE.

Károly, A. I., Fullér, R. & Galambos, P., 2018. Unsupervised clustering for deep learning: A tutorial survey.. *Acta Polytechnica Hungarica,* 15(8), pp. 29-53.

Keskar, N. S. & Socher, R., 2017. Improving Generalization Performance by Switching from Adam to SGD. *arXiv.*

Kingma, D. P. & Ba, J., 2014. Adam: A Method for Stochastic Optimization. *arXiv.*

Krizhevsky, A., Sutskever, I. & Hinton, G. E., 2012. *Imagenet classification with deep convolutional neural networks.* s.l., Advances in neural information processing systems.

Krsnik, I. et al., 2020. Automatic Annotation of Narrative Radiology Reports. *Diagnostics (Basel),* 10(4), p. 196.

Lau, J. H. & Baldwin, T., 2016. *An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. In Proceedings of the 1st Workshop on Representation Learning for NLP.* Berlin, Association for Computational Linguistics.

leCun, Y. A., Bottou, L., Orr, G. B. & Müller, K.-R., 2012. Efficient BackProp. In: *Montavon, G., Orr, G.B., Müller, KR. (eds) Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science.* Berlin, Heidelberg: Springer, pp. 9-48.

Lee, D. L., Chang, H. & Seamons, K., 1997. Document ranking and the vector-space model. *IEEE Software,* 14(2), pp. 67-75.

Lee, J. et al., 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics,* 36(4), pp. 1234-1240.

Lee, S. H., Maenner, M. J. & Heilig, C. M., 2019. A comparison of machine learning algorithms for the surveillance of autism spectrum disorder. *PLOS One.*

Le, Q. & Mikolov, T., 2014. *Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on International Conference on Machine Learning.* s.l., JMLR.org.

Li, T., Mei, T., Kweon, I.-S. & Hua, X.-S., 2010. Contextual Bag-of-Words for Visual Categorization. *IEEE Transactions on Circuits and Systems for Video Technology,* pp. 381-392.

Liu, D., Li, Y. & Thomas, M. A., 2017. *A Roadmap for Natural Language Processing Research in Information Systems.* Hilton Waikoloa Village, s.n.

Liu, Y. et al., 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv.*

Liu, Y. et al., 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv.*

Lloyd, S. P., 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory,* 28(2), pp. 129-137.

Lovins, J. B., 1968. Development of a stemming algorithm. *Mechanical Translation & Computational Linguistics,* pp. 22-31.

Lundberg, S. M. & Lee, S.-I., 2017. A Unified Approach to Interpreting Model Predictions. In: I. G. a. U. V. L. a. S. B. a. H. W. a. R. F. a. S. V. a. R. Garnett, ed. *Advances in Neural Information Processing Systems 30.* s.l.:Curran Associates, Inc., pp. 4765-4774.

Luong, M.-T., Pham, H. & Manning, C. D., 2015. Effective Approaches to Attention-based Neural Machine Translation. *arXiv.*

Malouf, R., 2002. *A comparison of algorithms for maximum entropy parameter estimation.* s.l., COLING-02: proceedings of the 6th conference on Natural language learning.

Malouf, R., 2002. A comparison of algorithms for maximum entropy parameter estimation. *COLING-02: proceedings of the 6th conference on Natural language learning,* Volume 20, pp. 1-7.

Manning, C. D. et al., 2014. *The Stanford CoreNLP Natural Language Processing Toolkit.* s.l., Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations.

Marcus, M., Santorini, B. & Marcinkiewicz, M. A., 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics,* pp. 313-330.

Matthews, B. W., 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure,* 405(2), pp. 442-451.

McCormick, P. J., Elhadad, N. & Stetson, P. D., 2008. *Use of Semantic Features to Classify Patient Smoking Status.* s.l., s.n., p. 450–454.

McKinney, W., 2010. *Data Structures for Statistical Computing in Python.* s.l., s.n., pp. 51-56.

Merck Sharp & Dohme, 2021. *MSD Manual Consumer Version.* [Online]
Available at: https://www.msdmanuals.com/
[Accessed 29 November 2021].

Meyer, D., Leisch, F. & Hornik, K., 2003. The support vector machine under test. *Neurocomputing,* 55(1-2), pp. 169-186.

Mikolov, T., Chen, K., Corrado, G. & Dean, J., 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv.*

Mikolov, T. et al., 2018. *Advances in Pre-Training Distributed Word Representations. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).* Miyazaki, Japan, European Language Resources Association (ELRA).

Molnar, C., 2019. *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable.* s.l.:s.n.

Monea, A. M. & Marginean, A. N., 2021. Medical Question Entailment based on Textual Inference and Fine-tuned BioMed-RoBERTa. *in: IEEE 17th International Conference on Intelligent Computer Communication and Processing (ICCP).*

Munkhdalai, T., Liu, F. & Yu, H., 2018. Clinical Relation Extraction Toward Drug Safety Surveillance Using Electronic Health Record Narratives: Classical Learning Versus Deep Learning. *JMIR Public Health and Surveillance.*

Nair, V. & Hinton, G. E., 2010. *Rectified linear units improve restricted boltzmann machines.* s.l., ICML'10: Proceedings of the 27th International Conference on International Conference on Machine Learning.

National Health Service, 2019. *Stopped or missed periods.* [Online]
Available at: https://www.nhs.uk/conditions/stopped-or-missed-periods/
[Accessed 05 06 2022].

NHS Wales, 2017. *Rules for managing referral to treatment waiting times,* s.l.: wales.nhs.uk.

NHS Wales, 2020. *NHS Wales Data Dictionary.* [Online]
Available at:
http://www.datadictionary.wales.nhs.uk/index.html#!WordDocuments/mainspecialtyconsultant.htm

Opitz, J. & Burst, S., 2019. Macro F1 and Macro F1. *arXiv.*

Patterson, O. & Hurdle, J. F., 2011. *Document clustering of clinical narratives: a systematic study of clinical sublanguages. In AMIA Annual Symposium Proceedings (Vol. 2011, p. 1099).* s.l., American Medical Informatics Association.

Pedregosa, F. et al., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12,* pp. 2825-2830.

Pennington, J., Socher, R. & Manning, C. D., 2014. *GloVe: Global Vectors for Word Representation.* Doha, Qatar, Association for Computational Linguistics, pp. 1532-1543.

Platt, J., 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers,* 10(3), pp. 61-74.

Porter, M. F., 1980. An algorithm for suffix stripping. *Program,* pp. 130-137.

Porter, M. F., 2001. Snowball: A language for stemming algorithms. *published online.*

Qian, N., 1999. On the momentum term in gradient descent learning algorithms. *Neural Networks: The Official Journal of the International Neural Network Society,* 12(1), pp. 145-151.

Rabhi, S., Jakubowicz, J. & Metzger, M.-H., 2019. Deep Learning versus Conventional Machine Learning for Detection of Healthcare-Associated Infections in French Clinical Narratives. *Methods Inf Med,* 58(1), pp. 31-41.

Rajapaske, T. C., 2019. *Simple Transformers.* [Online]
Available at: https://github.com/ThilinaRajapakse/simpletransformers

Rajendran, S. & Topaloglu, U., 2020. Extracting Smoking Status from Electronic Health Records Using NLP and Deep Learning. *AMIA Jt Summits Transl Sci Proc.,* pp. 507-516.

Reiter, E. & Dale, R., 1997. Building applied natural language generation systems. *Natural Language Engineering.*

Ribeiro, M. T., Singh, S. & Guestrin, C., 2016. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier.* San Francisco, CA, USA, In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Rong, X., 2016. word2vec Parameter Learning Explained. *arXiv.*

Rosenblatt, F., 1958. The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain. *Psychological Review,* pp. 65-386.

Salton, G., Wong, A. & Yang, C., 1975. *A Vector Space Model for Automatic Indexing.* New York, NY, USA: Association for Computing Machinery.

Sarwar, A. et al., 2019. A novel method for content-based image retrieval to improve the effectiveness of the bag-of-words model using a support vector machine. *Journal of Information Science,* pp. 117-135.

Savova, G. K. et al., 2010. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *Savova, G. K., Masanz, J. J., Ogren, P. V., Zheng, J., Sohn, S., Kipper-Schuler, K. C., & Chute, C. G. (2010). Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. Journal of the Americ,* 17(5), pp. 507-513.

Semaan, P., 2012. Natural Language Generation: An Overview. *Journal of Computer Science & Research (JCSCR),* 1(3), pp. 50-57.

Sevenster, M. et al., 2015. Natural Language Processing Techniques for Extracting and Categorizing Finding Measurements in Narrative Radiology Reports. *Applied Clinical Informatics,* pp. 600-610.

Smith, L. N., 2017. *Cyclical Learning Rates for Training Neural Networks.* s.l., 2017 IEEE Winter Conference on Applications of Computer Vision (WACV).

Snomed International, 2022. *Snomed-CT: 5-step briefing.* [Online]
Available at: https://www.snomed.org/snomed-ct/five-step-briefing
[Accessed 12 12 2022].

Sokolova, M. & Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. *Information Processing & Management,* 45(4), pp. 427-437.

Soldaini, L. & Goharian, N., 2016. QuickUML: a fast, unsupervised approach for medical concept extraction.

Spark Jones, K., 1972. A STATISTICAL INTERPRETATION OF TERM SPECIFICITY AND ITS APPLICATION IN RETRIEVAL. *Journal of Documentation,* pp. 11-21.

Spasic, I., Ananiadou, S., McNaught, J. & Kumar, A., 2005. Text mining and ontologies in biomedicine: making sense of raw text. *Brief Bioinform. ,* 6(3), pp. 239-251.

Spasic, I. & Lovis, C., 2020. Patient Triage by Topic Modeling of Referral Letters: Feasibility Study. *JMIR Medical Informatics,* 8(11).

Srivastava, N. et al., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *JLMR,* 15(56), pp. 1929-1958.

Sun, W., Rumshisky, A. & Uzuner, O., 2013. Evaluating temporal relations in clinical text: 2012 i2b2 Challenge.. *Journal of the American Medical Informatics Association : JAMIA,* 20(5), pp. 806-813.

Suominen, H. et al., 2013. *Overview of the ShARe/CLEF eHealth Evaluation Lab 2013. In: Information Access Evaluation. Multilinguality, Multimodality, and Visualization. CLEF 2013. Lecture Notes in Computer Science.* s.l., Springer, Berlin, Heidelberg.

Surkova, E., Nikolayevskyy, V. & Drobniewski, F., 2020. False-positive COVID-19 results: hidden problems and costs. *The Lancet Respiratory Medicine,* 8(12), pp. 1167-1168.

Su, T. & Dy, J. G., 2007. In search of deterministic methods for initializing K-means and Gaussian mixture clustering. *Intelligent Data Analysis,* 1:11(4), pp. 319-38.

Tang, D., Qin, B. & Liu, T., 2015. *Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.* Lisbon, Association for Computational Linguistics.

Tynecki, P. et al., 2020. PhageAI - Bacteriophage Life Cycle Recognition with Machine Learning and Natural Language Processing. *BioRxiv.*

Uğuz, H., 2011. A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. *Knowledge-Based Systems,* 24(7), pp. 1024-1032.

Uzuner, Ö. et al., 2012. Evaluating the state of the art in coreference resolution for electronic medical records.. *Journal of the American Medical Informatics Association : JAMIA,* 19(5), pp. 786-791.

Uzuner, O., Goldstein, I., Luo, Y. & Kohane, I., 2007. Identifying patient smoking status from medical discharge records.. *Journal of the American Medical Informatics Association : JAMIA,* 15(1), pp. 14-24.

Uzuner, Ö., Goldstein, I., Luo, Y. & Kohane, I., 2008. Identifying Patient Smoking Status from Medical Discharge Records. *Journal of the American Medical Informatics Association,* pp. 14-24.

Uzuner, Ö., South, B. R., Shen, S. & DuVall, S. L., 2011. 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association : JAMIA,* 18(5), pp. 552-556.

van der Maaten, L. & Hinton, G., 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research,* Volume 9, pp. 2579-2607.

Vaswani, A. et al., 2017. Attention Is All You Need. *arXiv.*

Virtanen, P. et al., 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods, 17(3)*, pp. 261-272.

Wachter, R., 2015. *The Digital Doctor: Hope, Hype, and Harm at the Dawn of Medicine's Computer Age.* 1st ed. s.l.:McGraw Hill.

Wang, A. et al., 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. *arXiv*.

Wattenberg, M., Viégas, F. & Johnson, I., 2016. How to Use t-SNE Effectively. *Distill.*

Webster, J. J. & Kit, C., 1992. *Tokenization as the Initial Phase in NLP.* s.l., s.n.

Welsh Government, 2022. *Waiting well? The impact of the waiting times backlog on people in Wales,* s.l.: s.n.

Welsh Government, April, 2022. *Waiting well? The impact of the waiting times backlog on people in Wales,* s.l.: s.n.

Welsh NHS Confederation, 2020. *Transforming NHS Wales services through digital developments,* s.l.: The NHS Confederation.

Weng, W.-H.et al., 2017. Medical subdomain classification of clinical notes using a machine learning-based natural language processing approach. *BMC Medical Informatics and Decision Making.*

Werbos, P. J., 1990. *Backpropagation through time: what it does and how to do it.* s.l., in Proceedings of the IEEE.

Werbos, P. J., 1990. Backpropagation through time: what it does and how to do it. *in Proceedings of the IEEE,* 78(10), pp. 1550-1560.

Whetzel, P. L. et al., 2011. ioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. *Nucleic Acids Res.*

Wicentowski, R. & Sydes, M. R., 2008. Using implicit information to identify smoking status in smoke-blind medical discharge summaries. *J Am Med Inform Assoc.,* pp. 29-31.

Wolf, T. et al., 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv.*

World Health Organization, 2004. *ICD-10 : international statistical classification of diseases and related health problems,* s.l.: World Health Organization.

Wu, Y. et al., 2018. Clinical Named Entity Recognition Using Deep Learning Models. *AMIA Annu Symp Proc. ,* pp. 1812-1819.

Yang, Y. & Liu, X., 1999. *A re-examination of text categorization methods.* Berkeley, s.n.

Yasunaga, M., Leskovec, J. & Liang, P., 2022. *LinkBERT: Pretraining Language Models with Document Links.* s.l., arXiv.

Yuan, C. & Yang, H., 2019. Research on K-Value Selection Method of K-Means Clustering Algorithm. *Multidisciplinary Scientific Journal,* 2(2), pp. 226-235.

Zech, J. et al., 2018. Natural Language–based Machine Learning Models for the Annotation of Clinical Radiology Reports. *Radiology.*

Zeiler, M. D., 2012. ADADELTA: An Adaptive Learning Rate Method. *Arxiv.*

Zhang, E., Thurier, Q. & Boyle, L., 2018. Improving Clinical Named-Entity Recognition with Transfer Learning. *Stud Health Technol Inform.,* pp. 182-187.

Zhang, Y., Jin, R. & Zhou, Z.-H., 2010. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics,* pp. 43-52.

Zhang, Y. et al., 2021. Biomedical and clinical English model packages for the Stanza Python NLP library. *Journal of the American Medical Informatics Association,* 28(9), pp. 1892-1899.

Zhou, X., Zhang, X. & Hu, X., 2006. *MaxMatcher: Biological Concept Extraction Using Approximate Dictionary Lookup. In: Yang Q., Webb G. (eds) PRICAI 2006: Trends in Artificial Intelligence. PRICAI 2006. Lecture Notes in Computer Science.* s.l., s.n.

Zhu, X. & Ghahramani, Z., 2002. Learning from Labeled and Unlabeled Data with Label Propagation. *CMU CALD tech report.*

Appendix

Appendix 1 Presenting Complaints Dataset Event Frequency Table

Speciality	Frequency
General Medicine	236961
General Surgery	131112
Paediatrics	126321
Trauma & Orthopaedics	92147
Cardiology	37612
Gynaecology	34088
Accident & Emergency	30945
Geriatric Medicine	22050
Paediatric Surgery	21587
ENT	19965
Urology	17798
Respiratory Medicine	16107
Plastic Surgery	14398
Gastroenterology	10825
Oral Surgery	10038
Endocrinology	7484
Adult Mental Illness	4506
Nephrology	3898
Neurosurgery	3127
Anaesthetics	2924
Midwifery Service	2319
Obstetrics	2150
Cardiothoracic Surgery	1523
Clinical Haematology	1180
Ophthalmology	951
Infectious Diseases	780
Rehabilitation Service	691
Medical Oncology	644
Neurology	634
Diabetic Medicine	618
Old Age Psychiatry	509
Clinical Oncology (Radiotherapy)	451
Clinical Pharmacology	377
Palliative Medicine	377
GP Other than Maternity	348
Stroke Medicine	282
Paediatric Neurology	193
Burns Care	87
Rheumatology	86

Vascular Surgery	78
Dermatology	76
Paediatric Burns Care	59
Child & Adolescent Psychiatry	53
Colorectal Surgery	51
Paediatric Plastic Surgery	42
Paediatric Neurosurgery	29
Learning Disability	24
Orthodontics	24
Paediatric Cardiology	20
Paediatric Dentistry	19
Dental Medicine	16
Community Medicine	14
Critical Care Medicine	14
Forensic Psychiatry	14
Cardiac Surgery	12
Paediatric Intensive Care	12
Paediatric Trauma and Orthopaedics	12
Maxillo-Facial Surgery	11
Medical Microbiology	10
Neonatology	10
Restorative Dentistry	9
Breast Surgery	7
Paediatric ENT	6
Upper Gastrointestinal Surgery	6
Antenatal Clinic	5
Pain Management	5
Radiology	5
Nursing Activity	3
Physiotherapy	3
Postnatal Clinic	3
Paediatric Medical Oncology	2
Thoracic Surgery	2
Dietetics	1
Hepatobiliary & Pancreatic Surgery	1
Hepatology	1
Intermediate Care	1
Paediatric Gastroenterology	1
Paediatric Opthamology	1
Paediatric Respiratory Medicine	1
Spinal Injuries	1
Spinal Surgery Service	1
Well Babies	1

Appendix 2 Complaints Dataset Subset Frequency Table

Speciality	Frequency
General Medicine	32456
General Surgery	16971
Trauma & Orthopaedics	11282
Paediatrics	7247
Cardiology	4997
Gynaecology	4309
Accident & Emergency	3774
Geriatric Medicine	3140
ENT	2361
Urology	2306
Respiratory Medicine	2117
Gastroenterology	1548
Plastic Surgery	1322
Oral Surgery	1146
Endocrinology	1080
Adult Mental Illness	575
Nephrology	572
Neurosurgery	385
Anaesthetics	352
Obstetrics	255
Midwifery Service	247
Cardiothoracic Surgery	204
Paediatric Surgery	196
Clinical Haematology	166
Infectious Diseases	115
Rehabilitation Service	106
Ophthalmology	100
Medical Oncology	82
Old Age Psychiatry	79
Neurology	76
Diabetic Medicine	72
GP Other than Maternity	62
Palliative Medicine	62
Clinical Oncology (Radiotherapy)	54
Clinical Pharmacology	50
Stroke Medicine	31
Vascular Surgery	16
Paediatric Neurology	12
Rheumatology	11
Child & Adolescent Psychiatry	8
Dermatology	8
Learning Disability	5

Colorectal Surgery	4
Dental Medicine	3
Medical Microbiology	3
Orthodontics	3
Paediatric Neurosurgery	3
Paediatric Plastic Surgery	3
Restorative Dentistry	3
Community Medicine	2
Forensic Psychiatry	2
Nursing Activity	2
Paediatric Burns Care	2
Radiology	2
Antenatal Clinic	1
Breast Surgery	1
Burns Care	1
Critical Care Medicine	1
Maxillo-Facial Surgery	1
Paediatric Dentistry	1
Paediatric Intensive Care	1
Paediatric Trauma and Orthopaedics	1
Spinal Surgery Service	1
Upper Gastrointestinal Surgery	1

Appendix 3 Referral Dataset Predicted Clusters

Cluster Labels	
0	**1477**
Cardiology	1
Gastroenterology	4
General Surgery	1370
Gynaecology	5
Orthopaedic	1
Paediatrics	73
Pain Management	1
Physiotherapy	3
Urology	19
1	**35929**
Cardiology	667
Care of the elderly	177
Clinical Immunology	290
Clinical Neuro-physiology	8
Community Orthopaedic	339
Dermatology	2783
Dietetics	1440
Endocrinology	871
ENT	2747
Gastroenterology	1103
General Medicine	929
General Surgery	2733
Geriatric Medicine	341
Gynaecology	2209
Haematology	488
Nephrology	338
Neurology	1863
Ophthalmology	557
Oral/Maxilo	758
Orthopaedic	4092
Paediatrics	1801
Pain Management	468
Physiotherapy	1984
Rapid Diagnostic Center	29
Rehabilitation	609
Rheumatology	1797
Thoracic Medicine	676
Urology	3178
Vascular Surgery	654
2	**2145**
Care of the elderly	2
Dietetics	6

	Endocrinology	2
	ENT	3
	Gastroenterology	743
	General Medicine	2
	General Surgery	1313
	Gynaecology	25
	Haematology	3
	Orthopaedic	4
	Paediatrics	7
	Rapid Diagnostic Center	9
	Rehabilitation	4
	Rheumatology	2
	Thoracic Medicine	2
	Urology	18
3		**3869**
	Cardiology	1
	Care of the elderly	3
	Community Orthopaedic	828
	Dermatology	7
	Dietetics	19
	General Surgery	6
	Neurology	1
	Orthopaedic	1862
	Paediatrics	10
	Pain Management	9
	Physiotherapy	1074
	Rheumatology	47
	Vascular Surgery	2
4		**3024**
	Cardiology	48
	Care of the elderly	11
	Clinical Immunology	22
	Clinical Neuro-physiology	5
	Community Orthopaedic	32
	Dermatology	393
	Dietetics	29
	Endocrinology	45
	ENT	271
	Gastroenterology	160
	General Medicine	41
	General Surgery	366
	Geriatric Medicine	10
	Gynaecology	297
	Haematology	30
	Nephrology	22
	Neurology	97

Ophthalmology	62
Oral/Maxilo	34
Orthopaedic	316
Paediatrics	123
Pain Management	38
Physiotherapy	149
Rapid Diagnostic Center	5
Rehabilitation	16
Rheumatology	57
Thoracic Medicine	76
Urology	221
Vascular Surgery	48
5	**5644**
Dermatology	5124
ENT	86
Gastroenterology	1
General Surgery	54
Gynaecology	66
Neurology	1
Ophthalmology	77
Oral/Maxilo	155
Orthopaedic	19
Paediatrics	7
Rapid Diagnostic Center	1
Thoracic Medicine	8
Urology	45
6	**1232**
Dietetics	7
ENT	548
General Medicine	2
General Surgery	2
Gynaecology	1
Neurology	12
Paediatrics	142
Pain Management	1
Rheumatology	1
Thoracic Medicine	516
7	**4666**
Care of the elderly	3
Community Orthopaedic	1
Endocrinology	2
Gastroenterology	2
General Medicine	1
General Surgery	5
Gynaecology	45
Haematology	5

	Nephrology	28
	Paediatrics	70
	Pain Management	2
	Physiotherapy	4
	Rapid Diagnostic Center	5
	Urology	4493
8		**2579**
	Clinical Immunology	101
	Dermatology	2156
	Endocrinology	1
	Gastroenterology	2
	General Medicine	2
	General Surgery	27
	Gynaecology	6
	Haematology	3
	Nephrology	5
	Ophthalmology	1
	Oral/Maxilo	4
	Orthopaedic	3
	Paediatrics	220
	Physiotherapy	1
	Rapid Diagnostic Center	1
	Rheumatology	32
	Thoracic Medicine	4
	Urology	7
	Vascular Surgery	3
9		**221**
	Cardiology	18
	Community Orthopaedic	6
	Dermatology	22
	Dietetics	3
	ENT	21
	Gastroenterology	9
	General Medicine	4
	General Surgery	72
	Geriatric Medicine	3
	Gynaecology	11
	Haematology	4
	Nephrology	2
	Neurology	8
	Oral/Maxilo	1
	Orthopaedic	6
	Paediatrics	2
	Pain Management	1
	Physiotherapy	3
	Rehabilitation	2

Rheumatology	4
Thoracic Medicine	9
Urology	10
10	**910**
Cardiology	21
Dermatology	63
Dietetics	45
ENT	42
Gastroenterology	18
General Surgery	326
Geriatric Medicine	1
Gynaecology	85
Haematology	7
Neurology	1
Orthopaedic	149
Pain Management	12
Physiotherapy	45
Rheumatology	12
Thoracic Medicine	16
Urology	67
11	**1655**
Care of the elderly	1
Community Orthopaedic	16
Dermatology	2
Dietetics	6
Endocrinology	1
ENT	1
General Surgery	6
Geriatric Medicine	4
Gynaecology	2
Haematology	1
Neurology	1
Orthopaedic	1201
Paediatrics	1
Pain Management	12
Physiotherapy	387
Rapid Diagnostic Center	2
Rehabilitation	1
Rheumatology	8
Urology	2
12	**1068**
Cardiology	28
Clinical Immunology	9
Clinical Neuro-physiology	5
Community Orthopaedic	40
Dermatology	107

Dietetics	34
Endocrinology	1
ENT	96
Gastroenterology	61
General Medicine	31
General Surgery	165
Geriatric Medicine	10
Gynaecology	55
Haematology	9
Nephrology	4
Neurology	14
Oral/Maxilo	6
Orthopaedic	76
Paediatrics	54
Pain Management	18
Physiotherapy	94
Rehabilitation	17
Rheumatology	19
Thoracic Medicine	34
Urology	81
13	**4514**
Care of the elderly	1
Dermatology	15
Endocrinology	1
ENT	4303
Gastroenterology	1
General Surgery	3
Neurology	8
Oral/Maxilo	9
Orthopaedic	1
Paediatrics	172
14	**422**
Clinical Immunology	1
Dermatology	68
Endocrinology	5
ENT	56
Gastroenterology	14
General Surgery	81
Gynaecology	39
Haematology	2
Nephrology	2
Neurology	7
Ophthalmology	4
Oral/Maxilo	12
Orthopaedic	58
Paediatrics	5

Pain Management	4
Rapid Diagnostic Center	6
Rheumatology	3
Thoracic Medicine	5
Urology	40
Vascular Surgery	10
15	**1629**
Cardiology	38
Care of the elderly	12
Clinical Immunology	4
Clinical Neuro-physiology	9
Community Orthopaedic	12
Dermatology	248
Dietetics	5
Endocrinology	33
ENT	219
Gastroenterology	21
General Medicine	10
General Surgery	251
Geriatric Medicine	2
Gynaecology	187
Haematology	11
Nephrology	11
Neurology	17
Ophthalmology	7
Oral/Maxilo	19
Orthopaedic	182
Paediatrics	48
Pain Management	6
Physiotherapy	14
Rapid Diagnostic Center	4
Rehabilitation	1
Rheumatology	31
Thoracic Medicine	35
Urology	174
Vascular Surgery	18
16	**5855**
Dermatology	4
Endocrinology	1
ENT	2
Gastroenterology	1
General Medicine	1
General Surgery	30
Gynaecology	5354
Haematology	3
Paediatrics	6

Pain Management	5
Physiotherapy	242
Thoracic Medicine	1
Urology	205
17	**5071**
Dermatology	22
Dietetics	2
ENT	1
General Surgery	5023
Gynaecology	1
Haematology	1
Ophthalmology	1
Orthopaedic	3
Paediatrics	12
Pain Management	1
Rheumatology	2
Thoracic Medicine	1
Urology	1
18	**1825**
Care of the elderly	2
Clinical Immunology	6
Dermatology	2
ENT	1715
General Surgery	2
Neurology	3
Oral/Maxilo	1
Paediatrics	86
Thoracic Medicine	8
19	**1928**
Dermatology	1868
ENT	19
General Surgery	1
Ophthalmology	10
Oral/Maxilo	30
20	**4912**
Cardiology	2781
Care of the elderly	28
Clinical Immunology	1
Dermatology	4
Dietetics	5
Endocrinology	6
ENT	63
Gastroenterology	28
General Medicine	50
General Surgery	18
Geriatric Medicine	27

Gynaecology	1
Haematology	8
Nephrology	13
Neurology	26
Orthopaedic	17
Paediatrics	298
Pain Management	1
Physiotherapy	9
Rapid Diagnostic Center	8
Rehabilitation	2
Rheumatology	3
Thoracic Medicine	1495
Urology	7
Vascular Surgery	13
21	**1957**
Care of the elderly	2
Clinical Neuro-physiology	197
Community Orthopaedic	10
Dermatology	9
General Surgery	2
Neurology	76
Orthopaedic	1564
Paediatrics	1
Pain Management	2
Physiotherapy	57
Rehabilitation	1
Rheumatology	29
Vascular Surgery	7
22	**6676**
Cardiology	1
Care of the elderly	32
Clinical Immunology	6
Dermatology	5
Dietetics	66
Endocrinology	14
ENT	31
Gastroenterology	2982
General Medicine	22
General Surgery	2418
Geriatric Medicine	4
Gynaecology	259
Haematology	130
Nephrology	11
Neurology	1
Oral/Maxilo	2
Orthopaedic	2

Paediatrics	466
Pain Management	4
Physiotherapy	4
Rapid Diagnostic Center	69
Rehabilitation	3
Rheumatology	7
Thoracic Medicine	5
Urology	115
Vascular Surgery	17
23	**996**
Dermatology	989
ENT	2
General Surgery	1
Orthopaedic	2
Rapid Diagnostic Center	2
24	**1512**
Gastroenterology	1
General Surgery	1
Gynaecology	1505
Urology	5
25	**3926**
Care of the elderly	1
Clinical Neuro-physiology	1
Community Orthopaedic	443
Dermatology	1
Dietetics	15
Endocrinology	4
Gastroenterology	5
General Surgery	11
Geriatric Medicine	4
Gynaecology	28
Haematology	14
Neurology	33
Orthopaedic	1141
Paediatrics	8
Pain Management	213
Physiotherapy	1904
Rapid Diagnostic Center	11
Rheumatology	65
Thoracic Medicine	1
Urology	18
Vascular Surgery	5
26	**2200**
Cardiology	2
Community Orthopaedic	320
Dermatology	24

ENT	1
General Surgery	8
Neurology	3
Orthopaedic	709
Pain Management	9
Physiotherapy	1098
Rheumatology	22
Thoracic Medicine	4
27	**2438**
Clinical Immunology	3
Endocrinology	2
ENT	2302
Gastroenterology	54
General Medicine	1
General Surgery	17
Haematology	1
Oral/Maxilo	8
Paediatrics	34
Thoracic Medicine	16
28	**848**
Clinical Immunology	1
Dermatology	844
General Medicine	2
Gynaecology	1
Grand Total	**111128**

Appendix 4 Clusters on referral data after reducing the features to a vocabulary of 2500.

Clusters	
0	**4311**
Cardiology	172
Care of the elderly	9
Clinical Immunology	14
Clinical Neuro-physiology	4
Community Orthopaedic	59
Dermatology	329
Dietetics	60
Endocrinology	82
ENT	259
Gastroenterology	253
General Medicine	47
General Surgery	326
Geriatric Medicine	37
Gynaecology	394
Haematology	50
Nephrology	28
Neurology	249
Ophthalmology	139
Oral/Maxilo	51
Orthopaedic	481
Paediatrics	163
Pain Management	110
Physiotherapy	203
Rapid Diagnostic Center	1
Rehabilitation	26
Rheumatology	195
Thoracic Medicine	136
Urology	382
Vascular Surgery	52
1	**1569**
Cardiology	5
Dermatology	7
Dietetics	708
Endocrinology	411
ENT	1
Gastroenterology	6
General Medicine	313
General Surgery	7
Gynaecology	3
Haematology	1
Nephrology	58

	Neurology	2
	Ophthalmology	2
	Oral/Maxilo	1
	Orthopaedic	3
	Paediatrics	1
	Pain Management	2
	Rapid Diagnostic Center	2
	Rehabilitation	2
	Rheumatology	5
	Thoracic Medicine	2
	Urology	23
	Vascular Surgery	4
2		**1927**
	Care of the elderly	3
	General Surgery	1
	Gynaecology	14
	Nephrology	17
	Paediatrics	20
	Physiotherapy	1
	Rapid Diagnostic Center	1
	Thoracic Medicine	2
	Urology	1868
3		**3008**
	Cardiology	48
	Care of the elderly	10
	Clinical Immunology	23
	Clinical Neuro-physiology	5
	Community Orthopaedic	38
	Dermatology	448
	Dietetics	28
	Endocrinology	36
	ENT	242
	Gastroenterology	151
	General Medicine	37
	General Surgery	355
	Geriatric Medicine	10
	Gynaecology	287
	Haematology	26
	Nephrology	20
	Neurology	90
	Ophthalmology	58
	Oral/Maxilo	33
	Orthopaedic	355
	Paediatrics	119
	Pain Management	41
	Physiotherapy	164

	Rapid Diagnostic Center	7
	Rehabilitation	16
	Rheumatology	55
	Thoracic Medicine	76
	Urology	186
	Vascular Surgery	44
4		**5476**
	Dermatology	5
	ENT	3
	Gastroenterology	1
	General Surgery	33
	Gynaecology	5086
	Haematology	4
	Oral/Maxilo	1
	Paediatrics	3
	Pain Management	2
	Physiotherapy	220
	Thoracic Medicine	1
	Urology	117
5		**36140**
	Cardiology	342
	Care of the elderly	153
	Clinical Immunology	369
	Clinical Neuro-physiology	17
	Community Orthopaedic	633
	Dermatology	5914
	Dietetics	629
	Endocrinology	324
	ENT	2419
	Gastroenterology	262
	General Medicine	510
	General Surgery	2313
	Geriatric Medicine	289
	Gynaecology	1984
	Haematology	272
	Nephrology	55
	Neurology	1584
	Ophthalmology	420
	Oral/Maxilo	685
	Orthopaedic	6382
	Paediatrics	1607
	Pain Management	550
	Physiotherapy	3615
	Rapid Diagnostic Center	13
	Rehabilitation	574
	Rheumatology	1692

	Thoracic Medicine	304
	Urology	1634
	Vascular Surgery	595
6		**5256**
	Dermatology	30
	Dietetics	2
	General Surgery	5200
	Gynaecology	1
	Haematology	1
	Ophthalmology	1
	Oral/Maxilo	1
	Orthopaedic	3
	Paediatrics	12
	Pain Management	1
	Rheumatology	2
	Thoracic Medicine	1
	Urology	1
7		**778**
	Cardiology	5
	Care of the elderly	15
	Dermatology	2
	Dietetics	6
	Endocrinology	2
	ENT	7
	Gastroenterology	453
	General Medicine	5
	General Surgery	108
	Geriatric Medicine	1
	Gynaecology	67
	Haematology	72
	Nephrology	4
	Neurology	1
	Oral/Maxilo	1
	Paediatrics	6
	Rapid Diagnostic Center	2
	Rehabilitation	6
	Rheumatology	4
	Thoracic Medicine	2
	Urology	8
	Vascular Surgery	1
8		**2129**
	Cardiology	2
	Community Orthopaedic	317
	Dermatology	43
	ENT	1
	General Surgery	9

Neurology	1
Orthopaedic	690
Pain Management	10
Physiotherapy	1029
Rheumatology	23
Thoracic Medicine	4
9	**2290**
Care of the elderly	1
Clinical Immunology	3
Dermatology	4
ENT	2183
Gastroenterology	39
General Medicine	1
General Surgery	12
Geriatric Medicine	1
Oral/Maxilo	7
Paediatrics	30
Thoracic Medicine	9
10	**871**
Cardiology	512
Care of the elderly	10
Clinical Immunology	2
Dermatology	1
Dietetics	6
Endocrinology	2
ENT	10
Gastroenterology	4
General Medicine	4
General Surgery	1
Geriatric Medicine	3
Haematology	5
Nephrology	10
Neurology	2
Orthopaedic	1
Paediatrics	11
Rapid Diagnostic Center	1
Rehabilitation	2
Rheumatology	3
Thoracic Medicine	278
Vascular Surgery	3
11	**1466**
Cardiology	1
Gastroenterology	3
General Surgery	1360
Gynaecology	5
Orthopaedic	3

Paediatrics	71
Pain Management	1
Physiotherapy	3
Urology	19
12	**145**
Dermatology	22
Endocrinology	4
ENT	16
Gastroenterology	4
General Surgery	16
Gynaecology	17
Nephrology	1
Neurology	8
Ophthalmology	4
Oral/Maxilo	4
Orthopaedic	28
Paediatrics	4
Pain Management	1
Rheumatology	4
Thoracic Medicine	2
Urology	6
Vascular Surgery	4
13	**7916**
Dermatology	7349
ENT	96
Gastroenterology	1
General Surgery	52
Gynaecology	67
Neurology	1
Ophthalmology	88
Oral/Maxilo	186
Orthopaedic	25
Paediatrics	7
Thoracic Medicine	3
Urology	41
14	**1628**
Cardiology	38
Care of the elderly	12
Clinical Immunology	4
Clinical Neuro-physiology	9
Community Orthopaedic	12
Dermatology	248
Dietetics	5
Endocrinology	33
ENT	219
Gastroenterology	20

General Medicine	10
General Surgery	251
Geriatric Medicine	2
Gynaecology	187
Haematology	11
Nephrology	11
Neurology	17
Ophthalmology	7
Oral/Maxilo	19
Orthopaedic	182
Paediatrics	48
Pain Management	6
Physiotherapy	14
Rapid Diagnostic Center	4
Rehabilitation	1
Rheumatology	31
Thoracic Medicine	35
Urology	174
Vascular Surgery	18
15	**2552**
Dermatology	91
Endocrinology	1
ENT	2284
Gastroenterology	1
General Surgery	4
Neurology	3
Oral/Maxilo	15
Paediatrics	152
Rheumatology	1
16	**9349**
Cardiology	19
Care of the elderly	28
Clinical Immunology	11
Dermatology	23
Dietetics	174
Endocrinology	68
ENT	129
Gastroenterology	2935
General Medicine	74
General Surgery	1776
Geriatric Medicine	20
Gynaecology	1024
Haematology	247
Nephrology	222
Neurology	23
Oral/Maxilo	7

Orthopaedic	13
Paediatrics	641
Pain Management	6
Physiotherapy	17
Rapid Diagnostic Center	100
Rehabilitation	6
Rheumatology	25
Thoracic Medicine	20
Urology	1701
Vascular Surgery	40
17	**604**
Dermatology	4
General Surgery	3
Paediatrics	94
Urology	503
18	**1857**
Care of the elderly	2
Clinical Immunology	6
Dermatology	3
ENT	1742
General Surgery	2
Neurology	3
Oral/Maxilo	1
Paediatrics	93
Thoracic Medicine	5
19	**1946**
Endocrinology	1
Gastroenterology	2
General Surgery	3
Haematology	3
Nephrology	3
Orthopaedic	1
Pain Management	2
Urology	1929
Vascular Surgery	2
20	**2091**
Cardiology	1
Care of the elderly	2
Clinical Neuro-physiology	11
Community Orthopaedic	107
Dermatology	121
Dietetics	2
Endocrinology	16
ENT	455
General Medicine	4
General Surgery	101

Geriatric Medicine	4
Gynaecology	1
Haematology	9
Neurology	63
Oral/Maxilo	21
Orthopaedic	471
Paediatrics	37
Pain Management	46
Physiotherapy	585
Rheumatology	25
Thoracic Medicine	2
Urology	3
Vascular Surgery	4
21	**3987**
Cardiology	1
Care of the elderly	3
Community Orthopaedic	837
Dermatology	9
Dietetics	19
General Surgery	6
Neurology	1
Orthopaedic	1936
Paediatrics	10
Pain Management	10
Physiotherapy	1105
Rheumatology	48
Vascular Surgery	2
22	**966**
Gastroenterology	1
General Surgery	1
Gynaecology	958
Urology	6
23	**3274**
Care of the elderly	2
Dietetics	4
Endocrinology	1
ENT	3
Gastroenterology	977
General Medicine	2
General Surgery	2184
Gynaecology	30
Haematology	5
Orthopaedic	4
Paediatrics	25
Rapid Diagnostic Center	11
Rehabilitation	4

Rheumatology	2
Thoracic Medicine	1
Urology	19
24	**1067**
Cardiology	28
Clinical Immunology	9
Clinical Neuro-physiology	5
Community Orthopaedic	40
Dermatology	107
Dietetics	34
Endocrinology	1
ENT	96
Gastroenterology	61
General Medicine	31
General Surgery	163
Geriatric Medicine	10
Gynaecology	55
Haematology	9
Nephrology	4
Neurology	14
Oral/Maxilo	6
Orthopaedic	77
Paediatrics	54
Pain Management	18
Physiotherapy	94
Rehabilitation	17
Rheumatology	19
Thoracic Medicine	34
Urology	81
25	**3435**
Cardiology	2383
Care of the elderly	21
Dietetics	5
Endocrinology	7
ENT	130
Gastroenterology	10
General Medicine	35
General Surgery	6
Geriatric Medicine	27
Gynaecology	2
Haematology	1
Nephrology	2
Neurology	54
Orthopaedic	1
Paediatrics	251
Pain Management	1

Physiotherapy	3
Rehabilitation	2
Thoracic Medicine	489
Urology	2
Vascular Surgery	3
26	**981**
Clinical Neuro-physiology	174
Community Orthopaedic	2
Neurology	47
Orthopaedic	738
Physiotherapy	16
Rheumatology	3
Vascular Surgery	1
27	**2137**
Care of the elderly	1
ENT	2096
Neurology	7
Orthopaedic	1
Paediatrics	32
28	**1972**
Cardiology	49
Care of the elderly	3
Clinical Immunology	3
Community Orthopaedic	2
ENT	138
Gastroenterology	26
General Medicine	25
General Surgery	19
Geriatric Medicine	2
Haematology	4
Nephrology	1
Neurology	2
Orthopaedic	13
Paediatrics	155
Physiotherapy	3
Rapid Diagnostic Center	10
Rheumatology	4
Thoracic Medicine	1506
Urology	3
Vascular Surgery	4
Grand Total	**111128**

Appendix 5 Classification reports for stemming and lemmatising referrals

	precision	recall	f1-score	support
Cardiology	0.95	0.97	0.96	682
Care of the elderly	0.93	0.64	0.76	59
Clinical Immunology	0.9	0.81	0.85	88
Clinical neuro-physiology	0.87	0.75	0.8	44
Community Orthopaedic	0.77	0.53	0.63	393
Dermatology	0.96	0.98	0.97	3017
Dietetics	0.97	0.93	0.95	346
endocrinology	0.85	0.82	0.84	201
ENT	0.95	0.97	0.96	2551
Gastroenterology	0.83	0.88	0.86	987
General medicine	0.84	0.68	0.75	202
General Surgery	0.93	0.93	0.93	2875
Haematology (clinical)	0.93	0.87	0.9	146
Nephrology	0.96	0.87	0.92	87
Ophthalmology	0.9	0.87	0.89	126
Oral/Maxillo facial surgery	0.9	0.79	0.84	218
Orthopaedic	0.87	0.93	0.9	2327
Paediatrics	0.91	0.77	0.83	741
Pain Management	0.88	0.78	0.83	147
Physiotherapy	0.86	0.87	0.87	1348
Rapid diagnostic centre	0.93	0.45	0.61	31
Rehabilitation	0.93	0.9	0.92	128
Rheumatology	0.91	0.92	0.92	430
Thoracic medicine	0.94	0.96	0.95	594
Urology	0.95	0.98	0.97	1766
Vascular surgery	0.84	0.87	0.86	146
Gynaecology	0.97	0.98	0.97	2024
Neurology	0.89	0.87	0.88	441
Geriatric medicine	0.88	0.79	0.83	81
Accuracy			0.92	22226
Macro average	0.9	0.84	0.87	22226
Weighted average	0.92	0.92	0.92	22226

	precision	recall	f1-score	support
Cardiology	0.95	0.97	0.96	682
Care of the elderly	0.93	0.64	0.76	59
Clinical Immunology	0.89	0.84	0.87	88
Clinical neuro-physiology	0.86	0.7	0.78	44
Community Orthopaedic	0.76	0.53	0.62	393
Dermatology	0.96	0.97	0.97	3017
Dietetics	0.96	0.94	0.95	346
endocrinology	0.86	0.83	0.84	201
ENT	0.94	0.97	0.96	2551
Gastroenterology	0.84	0.88	0.86	987
General medicine	0.85	0.68	0.76	202
General Surgery	0.93	0.93	0.93	2875
Haematology (clinical)	0.92	0.89	0.91	146
Nephrology	0.96	0.86	0.91	87
Ophthalmology	0.89	0.89	0.89	126
Oral/Maxillo facial surgery	0.89	0.77	0.82	218
Orthopaedic	0.86	0.93	0.89	2327
Paediatrics	0.9	0.76	0.83	741
Pain Management	0.9	0.79	0.84	147
Physiotherapy	0.86	0.87	0.86	1348
Rapid diagnostic centre	1	0.39	0.56	31
Rehabilitation	0.93	0.9	0.92	128
Rheumatology	0.9	0.92	0.91	430
Thoracic medicine	0.94	0.97	0.95	594
Urology	0.95	0.98	0.96	1766
Vascular surgery	0.81	0.86	0.83	146
Gynaecology	0.97	0.97	0.97	2024
Neurology	0.9	0.86	0.88	441
Geriatric medicine	0.89	0.77	0.82	81
Accuracy			0.92	22226
Macro average	0.9	0.84	0.86	22226
Weighted average	0.92	0.92	0.92	22226