Engineering Mathematics

Technical Project

Multi-Label Classification Of Medical Texts

Author
Andrew Corrigan

Supervisors
Raul Santos-Rodriguez
Avon Huxor

Abstract

Coding in the social sciences refers to the process in which data is categorised to facilitate analysis. The LeDeR learning from death review specialises in the coding of death reports of people with learning disabilities to analyse potential shortcomings in the health care system. The review focuses on a section of the death report called the "pen portrait" an unstructured passage of text describing the patients life. A time consuming and challenging task. The following report proposes a semi-automated model where codes are suggested to the coders based on a automated multi-label classification of the unstructured text . This is done with the aim of decreasing the time taken to code a report whilst also increasing the accuracy of which the pen portrait is coded. The report has two main sections; first the development of a classifier and second the creation of an effective user interface to display the classifications to the coders. The data consists of 6500 pre-coded death reports used to train, verify and test the classifier.

Two vectorisation methods are compared for an SVM classifier, a Term Frequency - Inverse Document Frequency (TF-IDF) vectoriser and a word2vec vectoriser. Both methods meet the predicted performance requirements, the TF-IDF has an F1-weighted accuracy of 0.64 and word2vec 0.61, both over the 0.6 target. This is predicted over 60% of all tokens, meeting the

second project goal of model breadth. The accuracy difference between the two vectorisation methods was not large on average however certain class variation was large: Word2vec performed well on codes with vaguer sentiment whereas TF-IDF performed well on labels with specific associated medical language. Weaknesses of each vectoriser are identified and clear improvements are suggested. The second half of the project develops a basic User Interface (UI) that displays the predictions made by the classifier and the classifiers confidence, the UI works as a stand alone program that accepts text input and outputs categorised, tokenised text. Valid target performance measures were met suggesting that an automatic classifier was developed so that it can be deployed to meet LeDeR's specific needs and aid the process of coding medical documents by meeting all of the projects design goals.

Contents

1 Introduction

1.1 Motivation

Reports have found there to be a large and unexplained gap of fourteen to eighteen years in the life-expectancy of individuals who have learning disabilities and those who are not effected. Mencap estimates that 1200 people a year with learning disabilities are dying unnecessarily because of a deficiency of appropriate healthcare [1],[2].

This disturbingly large gap in life expectancy has gone largely unexplained by current medical discourse. Whilst learning disabilities can and do have an effect on the everyday living experience of people and their social interactions, it should not have a negative effect on their medical well being. This has led to programmes such as the Learning Difficulties Mortality Review (LeDeR), being created. LeDeR is one of many NHS funded Learning from Deaths reviews across the UK that seek to create a deeper understanding of these fatalities and prevent them from occurring in the future, with others focusing on child deaths and deaths in hospitals. The programme was commissioned in response to the Confidential Inquiry into the Premature deaths Of people with Learning Disabilities (CIPOLD). The review has two main aims:

- To support improvements in the quality of health and social care service delivery for people with learning disabilities.
- To help reduce premature mortality and health inequalities for people with learning disabilities.

To achieve these goals the LeDeR programme collates death reports into a comprehensive database, a review is submitted to the database for each recorded death of a person with a learning disability. The LeDeR program will then review the report of the death and categorise and flag any important information to allow for a systematic approach to identifying features that could explain this gap in life expectancy. The reports are then submitted to this private online database to protect the identity and personal details of these vulnerable individuals. These reports can then be used to infer common reasons for death or poor practises by comparing the flags made by the LeDeR programme across death reports. This is so that the NHS can be alerted and changes to medical policy can be introduced so that this life expectancy gap can be reduced.

These documents consist of a few pages of check-boxes followed by a written report, referred to as a pen portrait. The majority of the information garnered about the patient is in the pen portrait as unstructured text data. It is typically written by medical professionals who have been involved with the care of the patients. The pen portrait is largely unstructured but follows some basic guidelines, a sample report is located in the appendix, it includes the guidelines given to the authors of the pen portraits. All details in the sample report are fictitious as to protect real people's data.

The current methods employed by LeDeR rely on the processing of large pieces of text by human coders, the coding process is a well documented method within the social sciences [3]. From the Coding manual for qualitative researchers a code is "A code in qualitative inquiry is most often a word or short phrase that symbolically assigns a summative, salient, essence-capturing, and/or evocative attribute for a portion of language-based or visual data", in LeDeR's case the code is a medical label or indicator of quality of life. A coder will read through the pen portrait and assign sections with a predetermined code, currently numbering some 300 labels (although previously

there were 600+ labels). With some of the most frequent being; "Circumstances of death" and "Epilepsy". These labels range from medically specific such as "Clinical cardiovascular problem" to medically vague such as "Useful material without code". This brings with it a multitude of issues with how the text is categorised, the lack of standardisation makes the more vague codes difficult to compare across reports.

The dependence on human coders is the project's current biggest shortcoming. It severely limits the rate at which the reports can be analysed as it is time consuming, with each reviewer coding an average of four reports a day. It is also prone to suffering from the subjectivity of the coders, different coders may code the same sentence or paragraphs in differing ways. This can lead to issues when the reports are compared in the database. Valuable information could be lost.

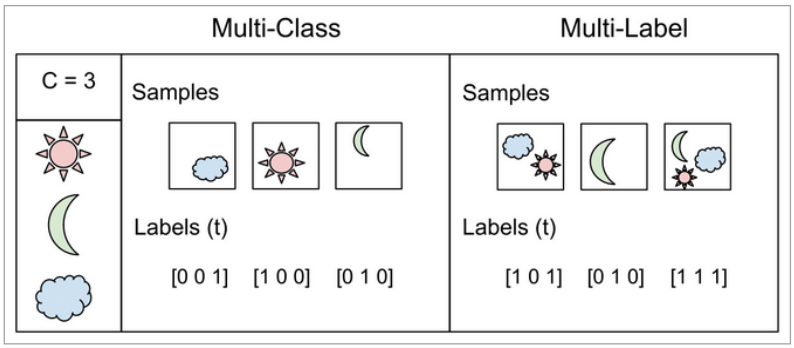
1.2 Goals

Due to the large amounts of unprocessed text data required to be coded by the LeDeR programme this project proposes a natural language processing (NLP) method to semi-automate the coding process, semi-automated in this case meaning that the coders would still make the final decision. This aids the human coders in categorising the text data as quickly and accurately as possible. This will enable reviews to processed faster and more consistently.

The proposed NLP program analyses text and suggest which labels are likely to the human coder. The coder can then confirm if the program is correct and select the appropriate label for submission. Therefore the process would be semi-automated. Due to the sensitive nature of the project and data used a fully automated process is inappropriate, any incorrect classification could have serious implications for the use of the labelled data. Therefore the final decision will remain with the human coders to mediate any mislabelled categorises.

This is a multi-label classification task, meaning that each text instance can belong to one or more classes. The distinction between multi-class and multi-label is shown in fig1. This is an important distinction to make, assigning the more common one label per item would result in incorrect labelling and a loss of information. The classifier also avoids making classifications between multiple classes with similar likelihoods. The coder can combine the program's suggestion and confidence with their own intuition to make the correct decision.

Figure 1: The distinction between multi-class and multi-label classification. Each shape represents a label that can assigned to a sample. Note that labels in multi-class classification are mutually exclusive where in multi-label they are not. [4]



The ultimate goal is to create an easy to use semi-automated coding tool for use by the LeDeR programme that reduces the time taken to code a report without compromising accuracy.

Whilst the process will still be time consuming compared to a fully automated model the supervised nature of the method has other advantages: A lower than normally acceptable accuracy compared to a fully automated model can be used as the classification will be verified by a human coder. The model will work by auto-coding the the tokens which will then be corrected by the coder. A classification accuracy of 50% is found by Gweon et al. to lead in a reduction of time [5] taken to classify documents compared to human coders alone. However this result is domain specific and would depend on the experience of the coder, the nature of the text being coded and the ease of use of the implementation. Conservatively a target accuracy of 60% should lead to a reduction in time in our application. Due to the domain specificity it is essential that the implementation of the classifier is tested to evaluate any effect on the time taken to code a document.

Whilst the accuracy may be less than accepted for a fully automated algorithm it is essential that the majority of the labels are able to be predicted by the semi-automated classification process. The breadth of the model is vital. The high number of codes that need to be remembered by the coders means that often recalling rarer codes is an issue. Not being able to recall codes could mean that important information contained within the death report is either given the wrong label or not labelled at all. This must at the very least not be encouraged by the program (by only suggesting the more popular codes) but should be discouraged, by also suggesting rarer codes as appropriate. The program will highlight this issue to the coders and help reveal previously unused correlations. The breadth of the model is just as important as the accuracy.

Goals of the Project

1. To generate a language model with at least 60% accuracy that can predict over 60% of the text required to be labelled

2. To have a language model tool that can be used by the current coders on their current workstations in the form of a user interface. It must be easy to use so that the time taken to code reports is reduced

3. To have a justifiable model, that makes it decision process and confidence level clear so that it can be questioned by the human coders and a reasoned outcome can be reached by the coder and classifier.

2 Data

The data that will be used to train the NLP model is a set of over 6500 death reports. These have already been processed, labelled and reviewed by the LeDeR program and as such will be used as a supervision training data-set. These reports consist of multiple choice questions and written responses. The appendix contains a full example report where the exact format can be seen. The information in this report is fabricated so that no sensitive information is shown.

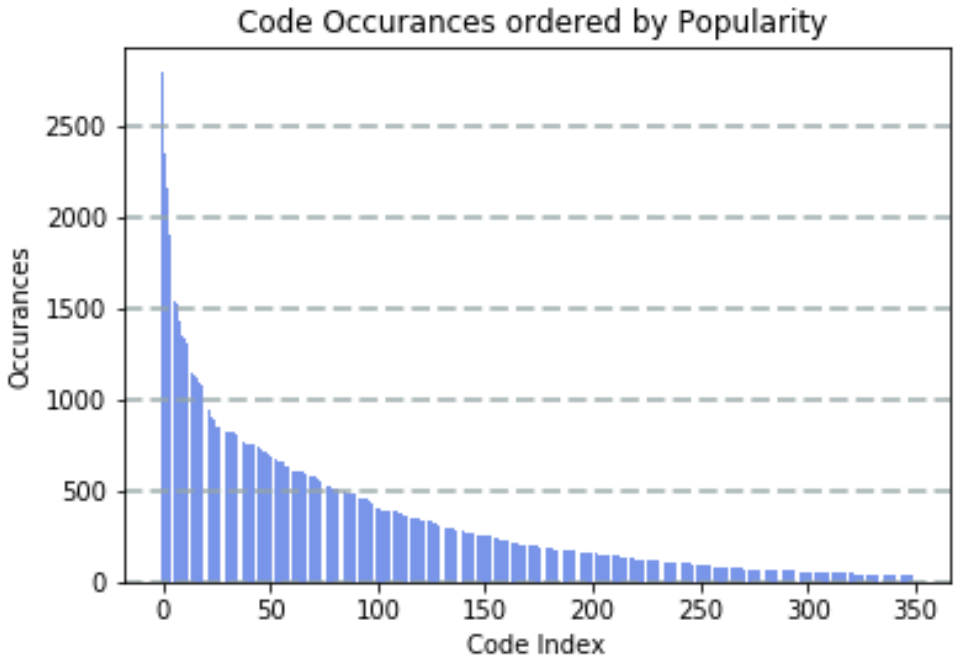
LeDeR focuses on the analysis the pen portrait, question 47 in the attached example report, figure **??**. It is essential for LeDeR to focus on the pen portrait as it is the most difficult section of the report to extract information from, often due to it's ambiguity. The review template gives some basic outline on how to structure the pen portrait and what information to include, despite this the pen portraits vary hugely both in content and length. Generally it is a description of the persons life and the circumstances leading up to their death, completed by a medical professional.

Currently when LeDeR is notified of a recent death the report will be submitted to the programme. The coder would then read through the portraits and assign each important piece of information with a corresponding label. At the time of writing there are roughly 300 codes that are used to label the portraits.

The current process is subject to a large amount of variance,pen portraits vary hugely in wording and content so regulations on how to label text for must reflect this. This coupled with variance in coders preferences result in an in-homogeneous data-set.

Codes greatly vary in the number of tokens assigned to each label. The counts of tokens per label range from some labels having few instances to some having over 3000. This range is demonstrated by figure 2 where the number of tokens per label is plotted. From the exponential

Figure 2: Code occurrences per code for the top 350 codes.Code index is the rank index ordered from code with the most occurrences to the least.



decline in occurrences of figure 2 as the code index increases there are a large number of labels with very few instances. The slope exhibits properties of an exponential decay.

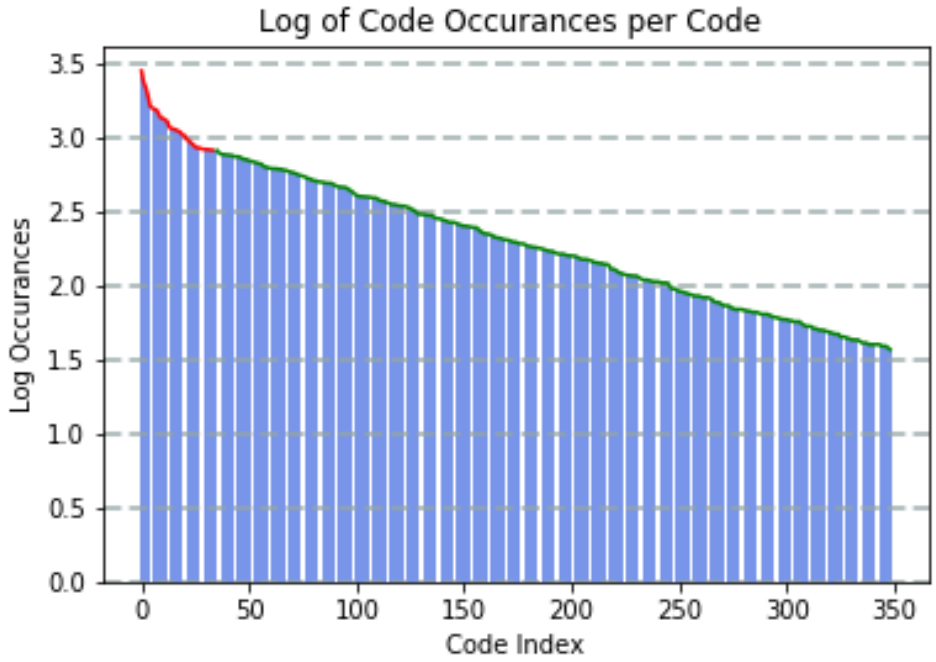
To examine the relationship further the log of code occurrences is plotted in figure 3. It shows two clear trends, a non-linear relationship between the code index and log occurrences for the fist 34 codes then a linear relationship for subsequent codes. The most popular codes follow one trend and the less popular codes another.

The two differences in trend can be highlighted by a change in colour, red representing the more "popular" codes and green representing the more "common" codes. As the log occurrences of the less popular codes ,highlighted in green, follow a roughly linear decline in relation to their popularity it can be said that it follows an exponential decay. Therefore where C is the label's popularity rank and N is the number of tokens per label the following relationship is implied for the more common codes:

$$N \propto e^{-mc}$$

Where m is the gradient of the decline.

Figure 3: Log of the number of Instances per code, the two trends of decay are highlighted in different colours. The popular code non-exponential decay in red, and the common codes exponential decay highlighted in green. The graph shows the two clear trends in decay of occurrences with the reduced popularity.



No such relationship can be established for the first 34 results.

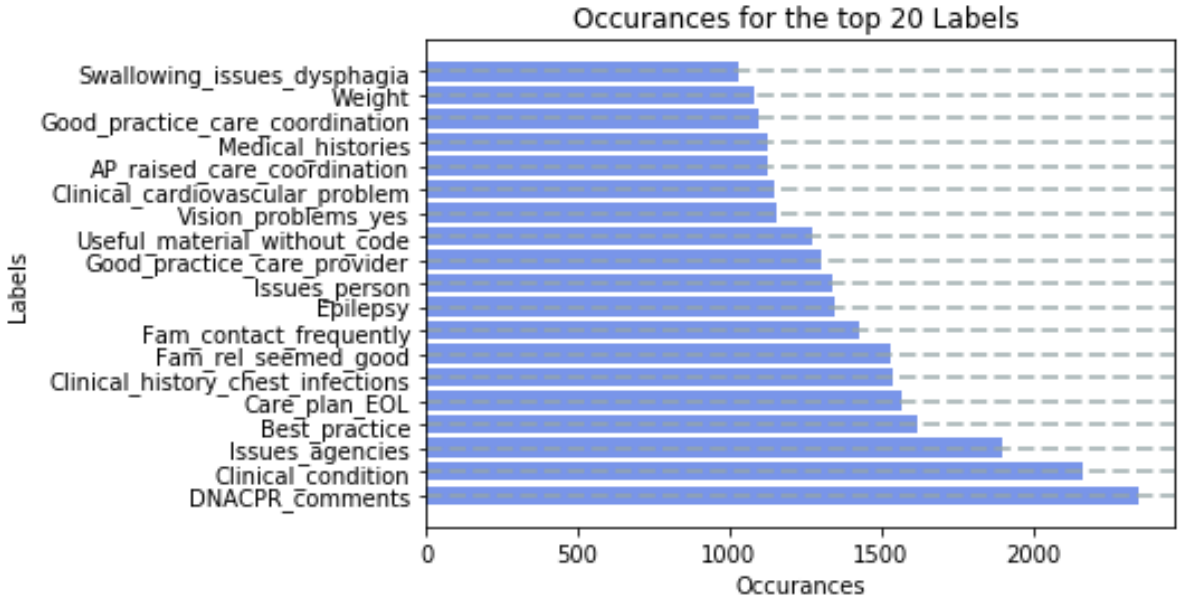
As the rank of the label count decreases the size of the training data for that specific label decreases exponentially. The proposed method prioritises predicting over the most popular codes first. This could prove problematic when adding less popular codes, there is the effect of diminishing returns. The less popular the code that is added the lower the increase in model coverage.

Figure 4 shows token occurrences for the top 20 labels . Showing not just the number of tokens per code but also the corresponding labels.

There is no clear relation amongst the most popular labels, the popular labels are both medically specific and vague. One of the most frequent labels with around 1000 instances is "useful material without code" a vague label, whereas "Clinical history of Chest infections" is specific. The specific labels would be less challenging to classify as there would often be specific language associated with chest conditions such as "asthma" and "coughing". Vaguer labels could be more challenging to classify, specifically with "Useful material without code" there could be no unifying content. The use of "Useful material without code" is justifiable in the context of a coder who wishes to mark something as useful without the knowledge of the specific code or a code existing for the content however training a classifier to recognise vague codes could produce unreliable predictions, it only has use as a tool for human coders. As the program is aiming to aid the coders in remembering a breadth of labels it would be more useful to suggest a less general and pre-existing code, if no code is still applicable the coder would still have the option to use the "useful content without code" label as the program will be semi supervised.

Not only does the token occurrences per label vary but the length of the tokens per label varies. Coders can label as long a piece of text as they seem fit, whilst this allows for versatility to cope

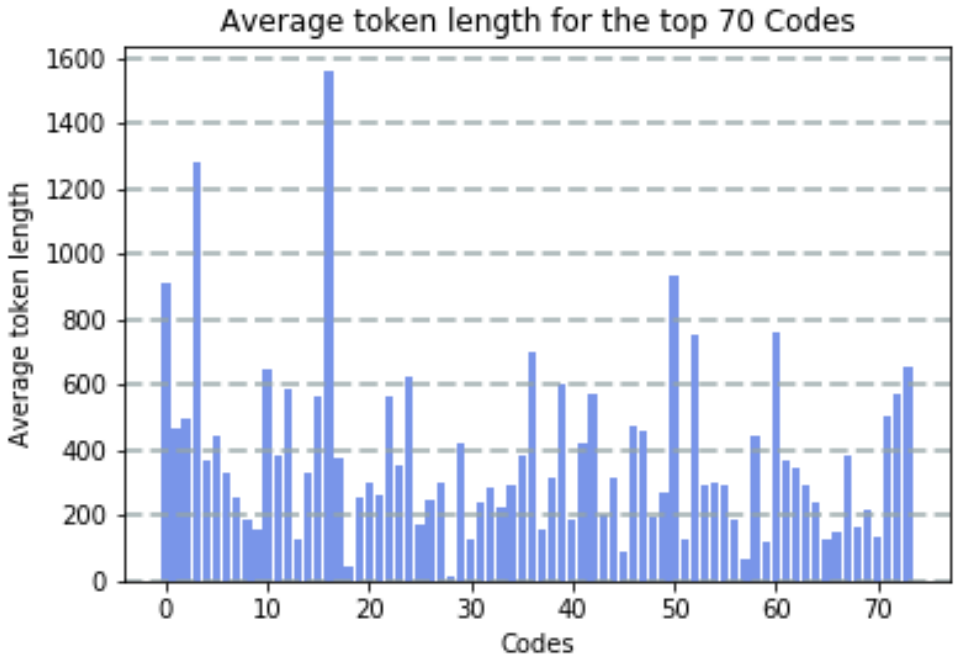
Figure 4: Number of occurrences per code including the associated code label, for the twenty most popular codes or for the code index [1-20].



for the variance in the pen portrait it could prove problematic for automation as information not relevant to the label could be included.

We show the average length of the token per class in figure 5.

Figure 5: Average length of token per code for code index [1-70]

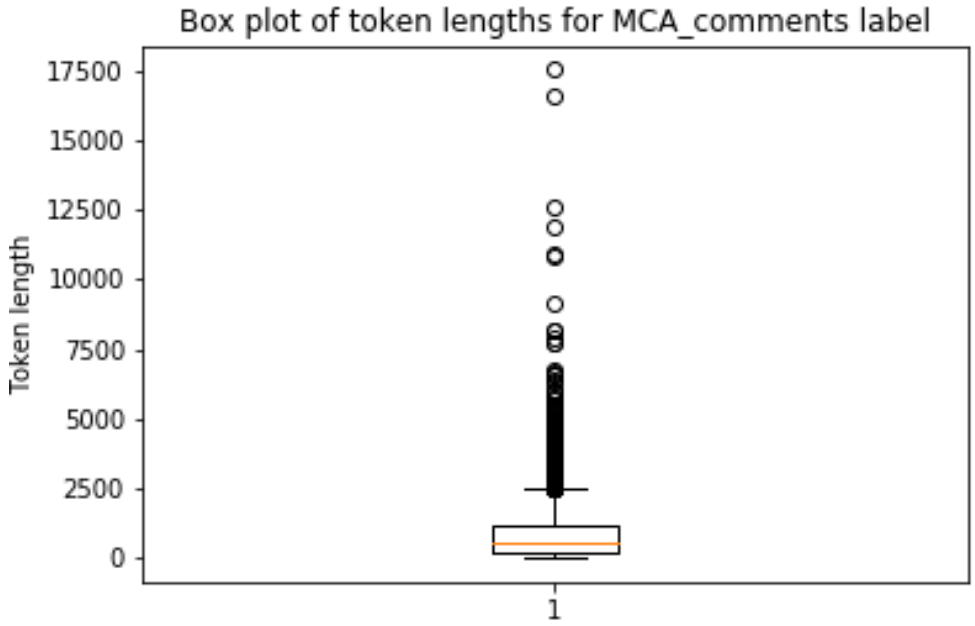


The average length varies hugely, some codes with over eight times the average length of others. This will effect the training of any algorithms. Some codes will have a combination of low

occurrences and low average token length meaning that the training algorithm will have very little to train itself on so predictions could be weakened.

The token length does not only vary from code to code but also varies within different codes. Here the box plot of the length of tokens for the most popular label, MCA comments [1], are shown 6 .

Figure 6: Box plot of the lengths of token assigned to the "MCA comments label". Code lengths are shown on the y axis.



The MCA comments code has a relatively tight spread of token length. With inter-quartile range being relatively small. However there are many outliers far from the average. One being 17,500 characters long. This shows that whilst there is a relatively tight spread there are still a large number of outliers that are far larger than the average.

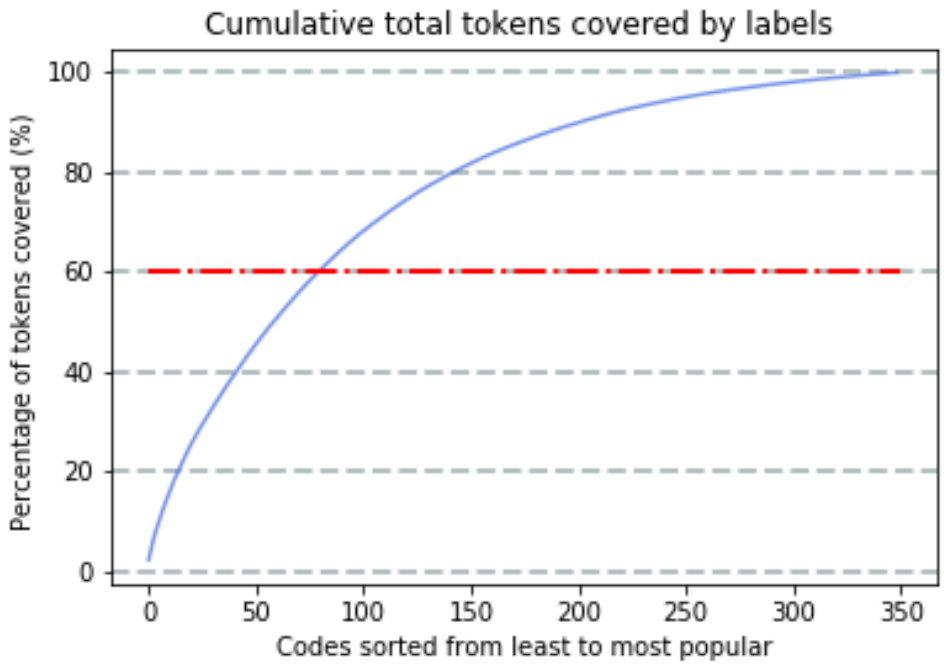
This is due to the differences in coders, some coders would label an entire document with one code if the entire document relates to the label. The entire document could also be labelled if there was an ongoing problem throughout the portrait. For example if a particular patient had a long term issue with their breathing that effected different areas of the patients life, the coder could label the entire report with "Clinical history of chest conditions". The reasoning for this would be that whilst not all information explicitly mentions chest conditions the entire report contributes to a patient living with a chest condition. The presence of these lengthened tokens can be problematic for two reasons. Classifying an entire document with one label does not exclude any language that is not explicitly relevant to the label, resulting in the classifier's input being skewed with irrelevant information. It is also problematic for the implementation of the classifier, it is difficult to detect if a document relates vaguely enough to a label to be classified as entirely belonging to the label without training a separate document level sentiment analysis.

The cumulative total of tokens covered is plotted in figure 7. The codes are plotted from least

[1]Mental Capacity Act http://www.legislation.gov.uk/ukpga/2005/9/contents

to most popular, so that 50 codes on the x-axis of the graph indicates how many tokens are covered by the 50 most popular codes.

Figure 7: Cumulative total of token coverage, codes added in order of popularity index. The red line shows the target coverage of 60%



From inspection it seems that roughly 75 codes should give us over 60% coverage.

Predicting over 75 codes exactly correlates to a coverage of 60.117% of the training dataset. This achieves the target coverage.

The reasoning to categorise the most popular codes first is so that the maximum coverage is achieved whilst training the minimum number of classifiers. The curve of the coverage shows that as the number of codes covered is increased the rate of coverage decreases. For example to increase the coverage from 60% to 80% would require double the number of classifiers.

2.1 Ethics

Medical data is unique in the scale and complexity of data collected. It offers insights unavailable in other areas of data mining due to the amount of information and the time over which it is collected. With medical data's uniqueness comes many unique challenges for the handling of it. The upmost care must be taken.

Historically there has been issues with medical data being collected then used for monitory gain, medical data is reported to have a value of \$361 per record compared to $\$1 - 2$ per stolen credit card on the black market[6]. Not only is the confidentiality of the data at risk of being breached but the physician-patient relationship is also at risk[7],[8].

Therefore medical data is anonymised by the LeDeR staff, this means removing all names from the document both the patients and all family members so that no medical details can be traced back to the patient by a third party. This however does not always happen, some names remain in the database meaning that the reports could be linked back to the original patients. Furthermore

some recent reports have even been able to link redacted medical data to the original patients. This means that the upmost care must be taken so that no data is lost and the security of the patients involved with the programme is not compromised.

To ensure this all of LeDeR's guidelines must be followed. The Bristol LeDer programme currently operates from a secure office, there are numerous security measures that must be followed. This secure office is the only place where the LeDeR data is accessible and must be kept this way. That means that the data should not be exported anywhere other than the confidential drive so that it cannot be accessed from the room. It is also important that no text data is outputted in the code as even the smallest amount of information could compromise the security of the data.

Only accuracy's and output using the example reports will be used so that the sensitive data is not compromised.

2.2 Data Pre-processing

Text data is fundamentally unstructured and as such can have a large amount of irrelevant information. Therefore before classification the irrelevant and potentially confusing information must be removed. This process is called pre-processing. Pre-processing can drastically increase the speed and accuracy of classification algorithms.
Here we show a fictitious sample report to demonstrate what pre-processing is necessary and how it will effect the text.

> Following a DoLS assessment, conditions were made that Matty should have a Care Act assessment with consideration of his need for support to access the community. Following that assessment in May 2016 additional funding was agreed for seven additional hours of 1:1 support a month. This enabled Matty to attend football matches and meet up with his old friends.

Removing numbers The first pre-processing step for our data is to remove any numbers. These occur frequently in the pen portraits in many forms; through prescription details, dates and ages. Dates such a "2016" would not be useful for the classification. Dates only provide information within the context of other dates. The correlation between time and label is subtle. This means that removing dates from the training corpus should increase the accuracy of the classifier.

Despite this numbers are often used as abbreviations for otherwise relevant text data. In the text example above "1:1" is used. One to one care is an indicator of the individual receiving close medical attention and as such is directly relevant to the quality of care label. Whilst relevant to the label and easy to detect there are often other indicators within the label. In this case "Additional" and "Support" are still indicators of quality of care meaning that the numbers can be removed without a loss of sentiment. Therefore removing numbers is a valid step as they can often be removed whilst maintaining the sentiment.

Changing all upper case. As upper and lower case letters are recorded as different characters algorithms would represent "dog" and "DOG" as having different meanings. This would result in a loss of information as the sentiment for the capitalised version and un-capitalised is the same.

This is especially prevalent in the medical domain due to the frequent use of improper nouns, which can often not be capitalised. This simple step replaces every uppercase character with a lowercase.

Removing stop words. These are divisions of natural language such as articles, prepositions and pro-nouns. e.g. "the","and","it". These carry information only in the context of other words and therefore can be disregarded for simpler NLP algorithms. This 'can also be covered by some vectorisation techniques such as Term Frequency - Inverse Document Frequency techniques, covered later in methods.

Lemmatising This reduces words to their roots so that for the purpose of the classifier "running" and "run" would have the same meaning. This however could not be useful depending on the corpus. For example "the boy's cars are different colors" would be lemmatised to "the boy car be differ color"

2.3 Training and Validation Sets

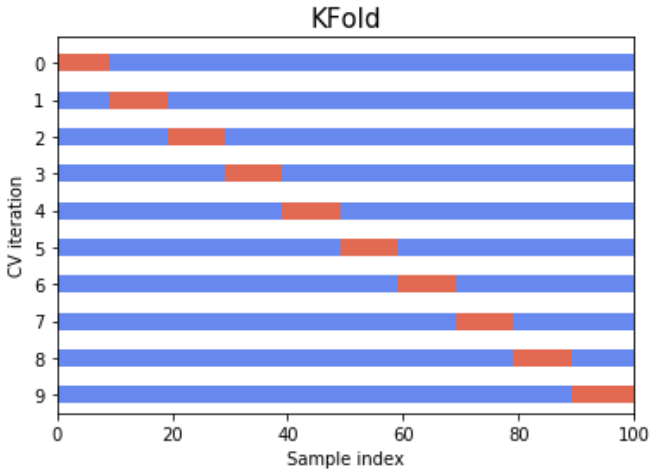
The LeDeR data is unique, there is not another data set with the same labels that can be used to test the accuracy of the predictions. We therefore must split the data into a training set, to train the classifiers. And a test set to test the accuracy of the learned models. The relationship between test and training is delicate it is possible when tuning the model using the test set for information to "leak" from the test to the training set thus biasing the performance of the test set so that it is not a true representation of the models performance. A third "validation" set is used to tune the hyper-parameters of the model by validating the models performance, this allows the test data to remain completely separate so that the hyper-parameters are not tuned to perform to the specific test data set but as a classifier on the data as a whole. This split however has it's disadvantages, as the LeDeR data-set is limited in size splitting the data into a usable set and two un-usable sets would reduce the already limited number of samples.

Instead we use a method called cross-validation. A set will still be held for final evaluation however the need for a completely independent validation set has been removed. The particular method is called K-fold validation. It works by selecting K-1 folds as the training set with the remaining fold as the validation. The model will then repeat this changing the validation set each time until the entire data set has been covered. The average will then be taken for all the data. This is shown in figure 8, for each fold or iteration the selected validation set is highlighted in red with the test data shown in blue. For each iteration a new fold is selected until the entire data-set is covered. It is clear from the diagram that every part of the test sample space has been selected for the validation set. This method is computationally expensive however minimises waste. A trade off that must be considered for each data-set.

.

Whilst cross validation removes the necessity for having a separate validation set the split between training and test data must be decided. Here we use a split of 70% for the training data and 30% for the test data. This is a larger than normal test set. This is as it is vital for the predictors to perform well on unseen samples.

Figure 8: The K-folds cross validation process, the red showing the selected validation set and the blue the training set for each CV iteration [9]



3 Prior Studies

Classification algorithms for natural language processing is a popular application of AI techniques and as such have been an area of large growth and documentation. It is important to understand the tested methods and their applicability to the LeDeR data set so that we can select the most appropriate method and develop a classifier to be as effective as possible. A systematic literature review of clinical coding and classification systems has been conducted by Stanfill et al. [10]. It was found that generally no method is more effective than another, performance is data specific. For that reason it is essential that when prior work is compared the domain of the application is compared to the domain in which we wish to apply the techniques.

3.1 SVM

Many papers have found support vector machines (SVMs) to achieve state of the art performance for NLP classification tasks[11–13]. The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. To separate the two classes of data points, there are many possible hyperplanes that could be chosen. The objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximising the margin distance provides some reinforcement so that future data points can be classified with more confidence.

One such paper by Nayak et al. [14]. demonstrates the effectiveness of SVMs. It compares SVMs to other classification methods for a multi-label classification task with multiple data-sets of different structures. They compare both the linear and RBF kernels to other current NLP classifiers such as a Naive Bayes classifier and Random Boltzman machines.

The main finding by Nayak et al. was that SVM's outperformed most other methods for the majority of the data sets. The F1 scores as calculated by Nayak et al. can be found in the following table. 9 It shows that the F1 score for both SVMs are consistently higher than other methods.

For the data sets with fewer features and more labels the SVM with an RBF kernel was found to be more effective. This is as the RBF kernel projects the vector onto a space of infinite dimensionality. Meaning that it is able to effectively predict across unseen samples. The LeDeR

Figure 9: Micro Precision, Recall and F1 of flat classification algorithms Nayak et al. [14]

Algorithm	Reuters	Citeseer	Bookmarks	Delicious	
	0.280	0.04	0.100	0.143	P
Naive Bayes	0.310	0.003	0.271	0.441	R
	0.294	0.005	0.146	0.215	F1
	0.404	0.515	0.426	0.405	P
SVM Kernel	0.419	0.383	0.212	0.279	R
	0.411	0.446	0.283	0.330	F1
	0.328	0.422	0.406	0.439	P
SVM rbf	0.356	0.349	0.274	0.278	R
	0.342	0.382	0.327	0.340	F1
	0.331	0.266	0.369	0.397	P
Random Forests	0.406	0.450	0.241	0.351	R
	0.365	0.335	0.292	0372	F1
	0.349	0.300	0.369	0.356	P
Extra Trees	0.395	0.458	0.248	0.356	R
	0.371	0.362	0.297	0.356	F1
	NA	NA	0.848	0.651	P
MLkNN	NA	NA	0.132	0.101	R
	NA	NA	0.228	0.175	F1
	NA	NA	0.010	0.118	P
BPML	NA	NA	0.919	0.727	R
	NA	NA	0.020	0.203	F1

data has a greater number of features and lower number of labels so 'Nayak's et al's result suggests that the linear kernel could perform better than the RBF. Despite this different kernels are easy to implement due to the robust nature of the model so kernels and hyper-parameters can be compared for our data-set to find the most effective combination.

Another study, by Madjarov et al. [15] confirms the findings by Nayak et al., Madjarov et al. test a variety of classification methods on datasets of different structures. They trial eight versions of SVM's, four types of decision trees and one K nearest neighbour on all of the data-sets. It is important that the data-set closest in structure to ours is compared so the results of the study are as relevant as possible.

Mediamill seems like a suitable comparison to our data-set as it has 101 labels, not dissimilar to the 75 that need to be predicted over to achieve target coverage of our dataset. It also has one on the larger numbers of training and test examples, similar to our large database of 6500 death reports. Another interesting comparison would be with the enron data-set, this has a similar number of labels at 53 but a much smaller number of training examples at 1123.

Mediamill achieved highest accuracy on the RF-PCT method, a version of a decision tree. It is worth noting that similar F1 scores were achieved for BR, CC and CLR all SVM adaptations. Enron achieved the highest F1 score for Homer, another adaptation of an SVM with other SVM methods F1 scores only marginally less. The F1 score for the majority of the data-sets might not be the highest for an SVM method but for data-sets in a similar domain to the LeDeR task all have high F1 scores for SVMs.

In addition to the results found by Nayak et al., Li et al. explore two key adaptations to SVM's for NLP tasks [16]. The first adaptation is the use of uneven margins. Li et al.adapt the SVM algorithm for a typically biased information extraction task. This allows adjustment to an SVM to negate the effect of small positive sample sizes. Li et al. show that it is beneficial to have a smaller margin for the smaller set of positive examples and a larger margin for the larger set of negative examples. This would help combat bias for unseen samples. SVMs naturally have a bias

Figure 10: Madjarov et al. selected data-sets and descriptions. Showing the number of training (tr.e.) and test (t.e.) examples, the number of features (D), the total number of labels (Q) and label cardinality (l_c) [15]

Dataset	*Domain*	*#tr.e.*	*#t.e.*	*D*	*Q*	l_c
emotions [31]	Multimedia	391	202	72	6	1.87
scene [32]	Multimedia	1211	1159	294	6	1.07
yeast [20]	Biology	1500	917	103	14	4.24
medical [16]	Text	645	333	1449	45	1.25
enron [33]	Text	1123	579	1001	53	3.38
corel5k [34]	Multimedia	4500	500	499	374	3.52
tmc2007 [35]	Text	21 519	7077	500	22	2.16
mediamill [36]	Multimedia	30 993	12 914	120	101	4.38
bibtex [37]	Text	4880	2515	1836	159	2.40
delicious [24]	Text	12 920	3185	500	983	19.02
bookmarks [37]	Text	60 000	27 856	2150	208	2.03

Figure 11: Madjarov et al. selected data-sets and F1 performance. DNF is where the model did not reach a classifier in the allocated time. [15]

Dataset	**BR**	**CC**	**CLR**	**QWML**	**HOMER**	**ML-C4.5**	**PCT**	**ML-*k*NN**	**RAkEL**	**ECC**	**RFML-C4.5**	**RF-PCT**
emotions	0.469	0.461	0.465	0.481	0.614	**0.651**	0.554	0.431	0.525	0.556	0.583	0.611
scene	0.714	0.742	0.713	0.710	0.745	0.587	0.551	0.658	0.754	**0.771**	0.395	0.553
yeast	0.650	0.657	0.655	0.654	**0.687**	0.614	0.578	0.628	0.661	0.670	0.589	0.614
medical	0.328	0.337	0.742	0.745	0.761	**0.768**	0.253	0.560	0.704	0.652	0.267	0.616
enron	0.582	0.484	0.600	0.525	**0.613**	0.546	0.295	0.445	0.564	0.602	0.505	0.552
corel5k	0.047	0.048	**0.293**	0.292	0.280	0.003	0.000	0.021	0.000	0.001	0.008	0.014
tmc2007	0.934	0.939	0.933	0.933	0.934	0.126	0.554	0.699	0.904	0.887	0.763	**0.948**
mediamill	0.557	0.539	0.134	0.135	0.579	0.054	0.490	0.570	0.471	0.483	0.572	**0.589**
bibtex	0.433	**0.434**	0.417	0.421	0.426	0.117	0.069	0.174	DNF	0.237	0.087	0.212
delicious	0.230	0.225	DNF	DNF	**0.343**	0.001	0.001	0.017	DNF	DNF	0.256	0.244
bookmarks	DNF	DNF	DNF	DNF	DNF	**0.257**	0.135	0.213	DNF	DNF	0.181	0.213

towards negative classification for smaller positive samples. This new technique could be useful for our data-set as there are 38 classes with only one positive result in the training set. This means that these labels would not be disregarded. The other development introduced by Li et al. is the "active learning" method (Note this is different from the conventional and previously discussed active learning). This finds the confidence the SVM has in it's predictions from the test set and finds the examples with the lowest confidence level. It will then introduce these examples into the training set to re-train the model. This works as the margins will be readjusted to incorporate the weaker predictions thus increasing the certainty of the model.

An SVM is therefore a reasonable starting point, they are easy to implement and have proven to be successful in domains similar to the LeDeR data. They also offer a large amount of adaptability with numerous optimisations that can be made within the SVM to improve the performance. They can easily be adapted to achieve the desired accuracy. Therefore the first method we will implement is an SVM and if the desired accuracy is not achieved further methods will be explored.

3.2 Neural Networks

Neural networks can potentially be more computationally expensive than SVMs but also have been shown to outperform SVMs for certain NLP applications. A neural network takes a large number of inputs and feeds these through a network of neurons that activate or deactivate depending on it's inputs and weights. These weights are then adjusted so that the network gives the desired output. Networks can be expanded to involve certain aspects like time dependencies in recurrent neural networks (RNN) or hierarchies by layering multiple neural networks. One such method explored by Tal Baume [**HA˙GRU**], is through the use of specialised neural networks. Baume et al. develop a model called Ha-GRU, it uses a series of gated recurrent neural networks coupled with a hierarchy. This dramatically reduces the computation time compared to a normal gated recurrent neural network (Hierarchical gated recurrent neural network) The HA-GRU works by identifying tokens relevant to each label and only analyses the token relevant to the label. This also makes the model more powerful as it uses a combination of token level and document level sentiment analysis. This could be useful compared to the more basic methods such as SVMs as it could solve the issues associated with models considering each word independently of the context of the sentence or document, often associated with such methods. Such as taking negations of a words to have the same meaning as the word. Baume compares the proposed HA-GRU method to an SVM, CBOW (continuous bag of words model) and convolutional neural network on both the MIMIC II and MIMIC III data-sets [2], consisting of roughly 44,000 discharge summaries [17]. The model assigns ICD9 [3] codes to the each discharge summary, of which there are roughly 1000 codes. This dataset is comparable to size and content of the LeDeR data set. Baume achieves an accuracy of 53.86% for the rolled up ICD9 codes on the MIMIC II data set. This is a significant improvement over the 32.5% accuracy achieved with the SVM method on the same data set. The only issue with this approach is that it applies labels to the discharge summary as a whole. The LeDeR requires a label estimation for each sentence meaning that the classification of the LeDeR is a much more complex task. This shows that a neural network approach may provide performance enhancements for classification tasks with a large number of labels despite being involved. The more involved nature of the neural network means that this would only be a valid development if a previous method does not achieve the target accuracy, the goal of the project is not to create the most accurate classifier possible but to improve the speed and accuracy of the LeDeR programme.

Yao et al. [18]use a novel approach to tackling the issue of labels with few training examples. They use a combination of a rule based method for labels withe few training examples where labels with larger numbers of training samples use a convolutional neural network (CNN)with word embeddings to identify the more populated labels. The results were compared to scikit's implementation of logistic regression and an SVM method with results being found to be significantly higher for the CNN than other tested methods.

This could be a useful development for the LeDeR data as many labels have a small numbers of training samples, an issue with methods such as SVMs. These labels with lower numbers of training samples also often feature label specific language meaning that a rule system could be effective.

[2] https://mimic.physionet.org/

[3] The ICD-9 coding system was used to code and classify mortality data from death certificates until 1999 in the US

3.3 Word Embeddings

Whilst the classification of the data is important there has also been developments in the processing of the text for input to the classifiers. These developments can have huge effects on the performance of the classifiers and can easily be implemented on top of existing classifiers to improve the performance. One such development is the use of word embeddings. Word embeddings work by calculating a vector representation of a word. They can then therefore manipulate words as if they were vectors, add and subtract words or more usefully in our case; calculate how similar one word is to another. A more complex approach than traditional methods. Techniques such as Word2Vec[4] learn similarities of words based on the context. For example "I had a great day" and "I had a good day" have similar meanings. Word embeddings are able to capture the similarity of words and represent this in vector form so that words with similar meanings are spatially close to each other. It is also much more effective than other NLP techniques at picking up meaning in the structure of the text as opposed to looking at the individual words. I.e; despite "Not fun" and "fun" having different meanings, an SVM would read them both as having the same. One such implementation is BioWord2Vec, [19] a specialised Word2Vec program for understanding biomedical language developed by Zhang et al... This method builds on the traditional word embedding models by not just taking each word as a vector but uses the sub-word information as well. This makes the method more robust and able to recognise out of vocabulary words. Something that could be quite common with the obscure language associated with medical applications. It demonstrates that the method performs particularly well on sentence similarity for clinical reports. It also demonstrates a higher precision than other word embedding models with the DDI extraction evaluation results. Recent developments in NLP are trending towards pre trained word embeddings. One such popular and controversial method is the BERT model [20]. BERT uses an extremely large but non specific corpus to train the word embeddings, one of the larger corpus's used is Wikipedia. This is useful for more general NLP tasks ,however due to the non-specific nature of the training corpus it does not perform well with specific applications[20]. This has since been improved though different, specialised versions of BERT being available. One such model is BioBert, it adds to the corpus of BERT with PubMed abstracts and full articles from PMC. This helps it understand the technical vocabulary used in biological and medical reports such as the LeDeR. Lee and Yoon's report [20] demonstrates the effectiveness of the pre-trained word embedding model. The BioBert model outperformed "state-of-the-art" models significantly on six of the nine tests. It also demonstrated that the larger the corpus the better the performance.

[4]https://code.google.com/archive/p/word2vec/

4 Classification

With prior work assessed for similar tasks an SVM is selected as the base classifier. Two vectorisation methods, a TF-IDF and word2vec are compared. The vectoriser with optimal performance for the LeDeR data-set will be selected and used in the tool to aid the coding process.

4.1 Methods

4.1.1 SVM

An SVM is a binary classifier, it works by finding a hyperplane that best classifies a pre-labelled data set. It does this by maximising the margin between the positively labelled training data and the negatively labelled training data. For a data-set in the form:

$$(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n) \tag{1}$$

Where $vecx_1$ is the p dimensional real vector of the data with y_1 being the corresponding label vector. In the binary case the label is either a -1 or 1.

A hyperplane can be written as the set of points $\vec{x}$ satisfying:

$$\vec{w} \cdot \vec{x} - b = 0 \tag{2}$$

Therefore data-points can be classified in the following way:

$$\vec{w} \cdot \vec{x} - b = 1 \tag{3}$$

Anything above the boundary is labelled with a 1. Therefore:

$$\vec{w} \cdot \vec{x} - b = -1 \tag{4}$$

Anything below the boundary is labelled with a -1.

For non-linearly separable data we introduce the hinge loss function.

$$\max\left(0, 1 - y_i(\vec{w} \cdot \vec{x}_i - b)\right) \tag{5}$$

This is a penalty function that decreases depending on the incorrectly classified data's distance from the margin. This allows for an SVM to train on and classify data that is non-linearly separable.

Therefore for a soft margin SVM we aim to minimise the following function.

$$\left[\frac{1}{n}\sum_{i=1}^{n} \max\left(0, 1 - y_i(\vec{w} \cdot \vec{x}_i - b)\right)\right] + \lambda\|\vec{w}\|^2 \tag{6}$$

λ is the trade off between increasing margin size, ensuring $\vec{x}_i$ lies on the correct side of the margin. $\lambda = 0$ will give a hard margin SVM.

As the TF-IDF (Term Frequency- Inverse Document Frequency vectorised LeDeR data has a dimension of 32,000 all visualisation of the SVM method will be done with an arbitrary two-dimensional data-set. The two class data is plotted in figure 12a, purple being one class, red the other. Figure 12b shows potential hyperplanes classifying the data with 100% accuracy. The hyperplane with optimal split must be found. A theoretical new datapoint is added, marked with

an X; this could be assigned different labels depending on which hyperplane is selected to classify the data, the choice of hyperplane has a huge effect on new data. In figure 12c the corresponding margin of error with each associated hyper plane is shown, the margin being the distance from the hyperplane to the closest training points. Minimising the previously mentioned function 6 selects the hyperplane with the largest decision margin, this will give the most robust model. The final figure 12d shows the optimal hyperplane with the largest margin. Several data points lie on the decision boundaries, these are the pivotal elements of the fit and are referred to as support vectors, adjusting these would adjust the fit. Due to the loss function any point further away from the boundary than the support vectors do not contribute to the fit. This insensitivity to distant points is a key strength of SVM's. This is relevant for our classifier as for each class there will be extremely large amounts of irrelevant data for each class, SVM's ability to disregard this irrelevant data is essential for classifier performance.

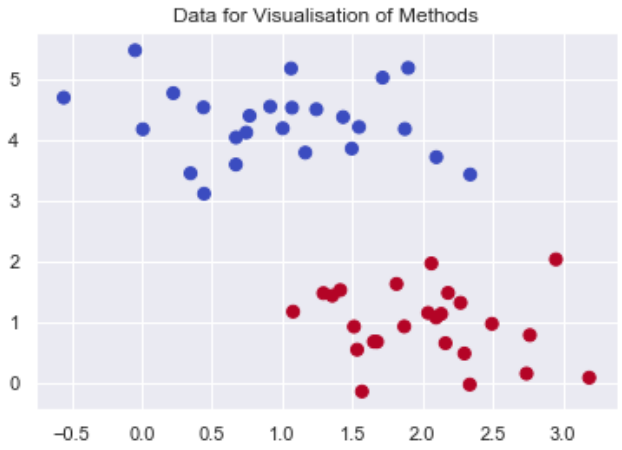
An SVM is only directly applicable to two class classification tasks. The LeDeR programme requires classification of roughly 70 classes so the SVM must be extended. To do this the most common method is to reduce the multi class classification problem to a series of two class classification problems. There are two commonly used approaches:

1. **One-versus-All** This creates a margin between the data for the selected class and all of the data for all other labels. New instances are predicted with a winner takes all approach, the new instance is assigned to the class with the highest output function. To ensure that all of the sample space is categorised it requires an extension of the value function to output continuous labels rather than binary labels.

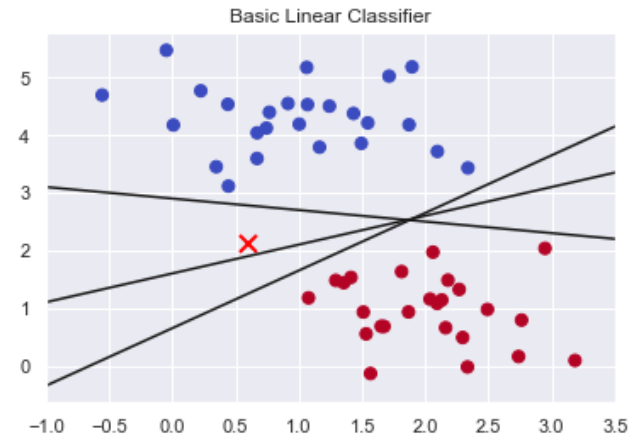
2. **One-versus-One** utilises a series of one vs one classification tasks. New instances are assigned by the max votes method. The new instance is tested for each one vs one classifier if a win is recorded for a class then the win count for the class is increased by one for that new instance. The class with the highest win count is then assigned to the new instance.

The two methods vary greatly in complexity. As the one versus all method trains one classifier for each class for N classes it will train N classifiers. Whereas for the one versus one method it must train a classifier for each possible pair of classes meaning for N classes it must train $\frac{N(N-1)}{2}$ classifiers. For 70 classes the one versus all would train 70 classifiers compared to $2,415$ classifiers required by one versus one. Despite the greatly increased computational complexity one versus one offers some advantages over one versus all, it is less sensitive to bias from imbalanced datasets. Due to the high number of classes and data-points we one versus all will be used as the increased complexity of one versus one could be too time consuming for the scope of the project.

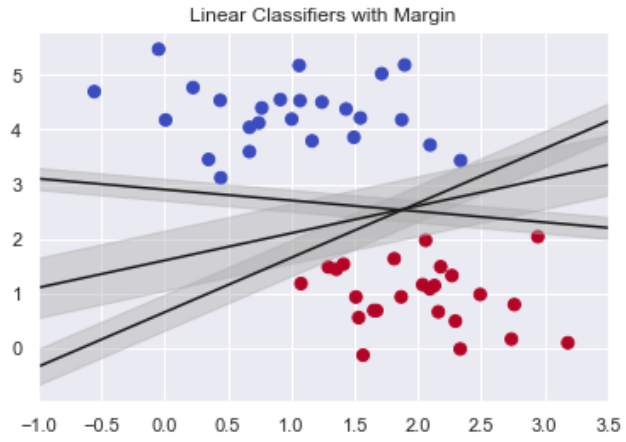
Our task is not only multi-class but also multi-label. This means that as well multiple classes labelling the data-set each individual data-point can have multiple labels. This means that the SVM must be extended further for both the training and testing of the SVM. One approach is to split the the problem into a series of binary classifier SVMs such that each label was represented by it's own independent classifier. This means the output would be a series binary values with 1 representing it having the label and -1 not. This is a simple extension of an SVM however it would be poor for the our application as a binary output would be difficult to interpret for the human coders, there is no measure of the model's confidence. It is essential that when can assigning codes the confidence the algorithm has in it's prediction is easily understood. This is so that the coders are able to make a decision if the label is correct. Instead a system is proposed using the previous extension from the one versus all multi class SVM. We use a normalised continuous



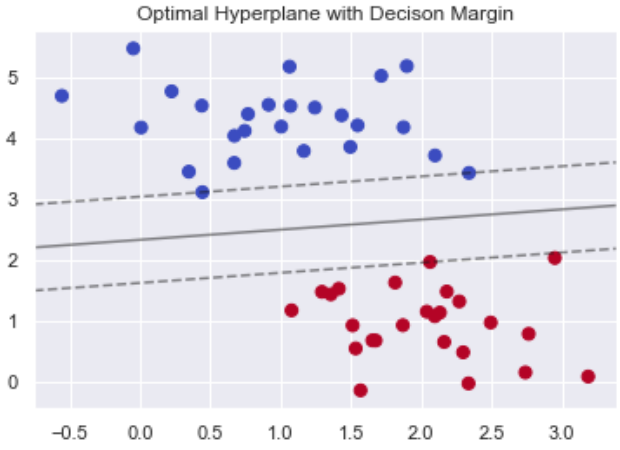
(a) The labelled data visualisation set.



(b) Data visualisation set with linear classifiers



(c) Linear separators with decision margins plotted



(d) Optimal hyperplane with decision boundaries

Figure 12: Plots showing the effect of hyper-plane suggestion on classifying a labelled arbitrary data set. [21]

output function to estimate the probability of a label being assigned to each new instance. This normalised probability will then be used to select the most relevant labels using a variable minimum threshold, the selected relevant labels can be displayed to the coder for verification. The model will be trained as if the data were not multi-label, if a single piece of text was assigned with multiple labels the text would be repeated in each label's training data set. This is the simplest way to train the model so that no information is lost.

With basic methodology decided we must optimise the SVM classifier to maximise the accuracy for the test data set. To do this we adjust the hyper-paramters using a grid search.

SVM's can be extended to use non-linear classifiers using the kernel trick. This works by mapping the data to a higher dimensional space where the data is linearly separable. Relevant dimensions of relatedness must be transferred to the higher dimension so that the model generalises well for unseen data. We do not want to overfit the model. To use an alternative kernel we simply replace the dot product mapping with a kernel of choice. Let

$$K(\vec{w}, \vec{x}_i) \tag{7}$$

be the kernel mapping such that for the dot product

$$K(\vec{w}, \vec{x}_i) = \vec{w} \cdot \vec{x}_i \tag{8}$$

As no one kernel performs better than another for all data-sets we must test multiple kernels to find the optimal classifier.

The linear kernel is simply:

$$K(\vec{w}, \vec{x}_i) = \vec{w} \cdot \vec{x}_i \tag{9}$$

The RBF kernel maps data such that:

$$k(\vec{x_i}, \vec{x_j}) = \exp(-\gamma \|\vec{x_i} - \vec{x_j}\|^2) \tag{10}$$

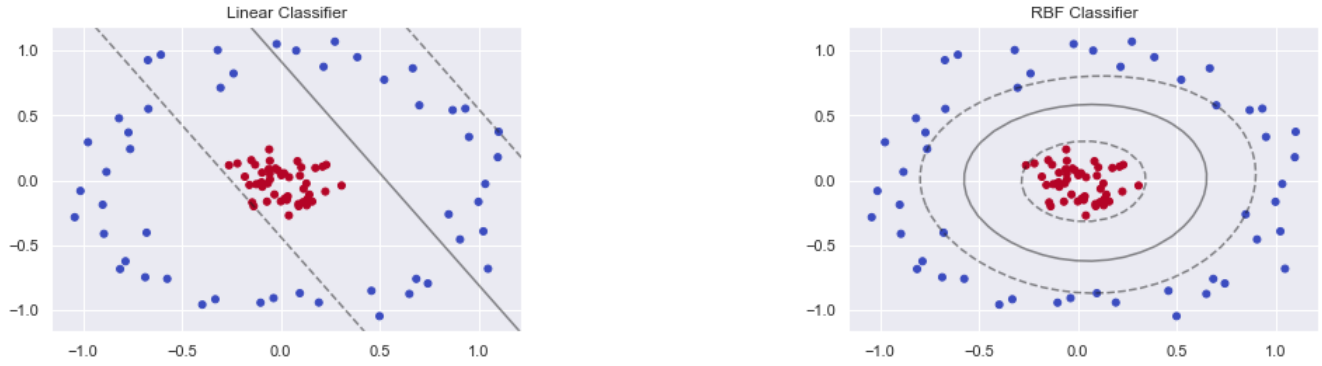
The RBF kernel works by mapping the data to an infinite dimensional Hilbert space (a vector space that is closed under the dot product).

The effects of the kernel on the decision boundary can be seen in figure13a and figure13b. The accuracy of the kernel is highly dependant on the structure of the data. As the text data is non-trivial to visulaise the optimal kernel must be found through the trial and error method of grid search.

As well as the kernel there are other hyperparamters to optimisie the performance of the SVM.

Different kernels have more hyperparameters however the cost value, referred to as C, is common to all kernels. This is the measure of how sensitive the cost function is to data-points beyond the margin, it influences λ in the soft margin minimissation function. The RBF kernel also has the γ parameter. This is a measure of how far the influence of a single training point reaches. The effects of varying and gamma are shown in figure 14.

. These must also be optimised to find the best performing classifier.

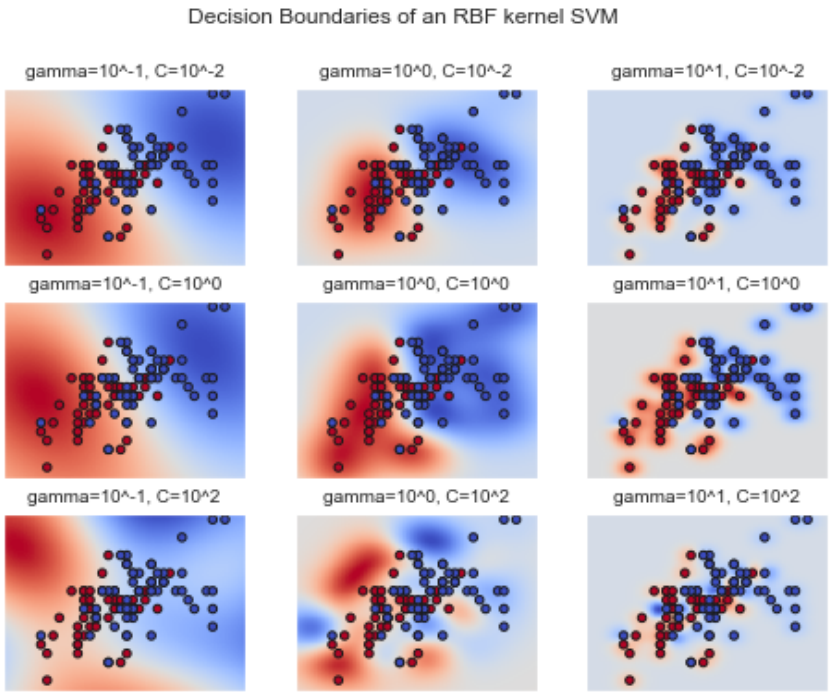


(a) SVM with Linear Classifier.

(b) SVM with RBF Kernel

Figure 13: Hyperplanes drawn with two different kernels, a linear and RBF kernel. The two classes of the arbitrary data points are highlighted in red and blue, note that the data's structure has a large effect on which kernel is effective.

Figure 14: The effect of varying γ and C for an RBF Kernel SVM



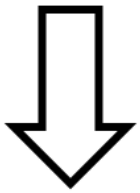
4.1.2 TF-IDF

Before applying and tuning an SVM we must convert the pre-processed text data into a numerical representation so that an SVM can separate it, it must be vectorised. There are numerous ways to do this with each having a large effect on the accuracy of the classifier. Whilst the SVM is generally accepted to achieve state of the art performance the vectoristation is effected largely by the data structure of the data and as such there is less of an industry standard approach. The first approach of two trialed is a one-hot encoding count vectoriser with a Term Frequency -Inverse

Document Frequency (TF-IDF) transform. A count vectoriser works by creating a vector that is dimensionally equal to the vocabulary required to be encoded. It then counts the number of times each word from the vocabulary occurs in the token that requires vectorisation and then generates a vector of the counts. This can be seen for a small vocabulary example in the following table 15.
.

Figure 15: The count vectorisation process for arbitrary sentences

Text
The patient suffered from a cough
XXX was suffering from a heart problem
His favourite haribo were heart shaped



Heart	cough	suffer
0	0	1
1	1	0
1	0	0

. This simple method therefore can create extremely large vectors depending on the size of the vocabulary that requires encoding. This means that for certain classification algorithms the large number of inputs can be problematic, it is easy to overfit the classifier. The other issue is that it weights each word equally, irrespective of it's contribution to the sentiment of the token. For example in a sentance such as "the cat" the words "the" and "cat" carry equal weighting despite "the" carrying little meaning.

We therefore use a transform to reduce the computational time used by large input vectors and over-fitting due to low sentiment words being equally weighted. The transform we will use is Term Frequency - Inverse Document Frequency (TF-IDF) transform. This is statistical measure that weights words on how important they are to the sentiment of the document. It reduces the influence words that carry little sentiment such as stop words. TF-IDF is a combination of two measures. Term frequency and Inverse document frequency. This are combined as follows. The score of the word t from the document d in the set of documents D is:

$$TFIDF(t, d, D) = tf(t, d) \cdot idf(t, D) \tag{11}$$

$$tf(t, d) = log[1 + freq(t, d)] \tag{12}$$

$$idf(t, D) = log(\frac{N}{count(d \in D : t \in d}) \tag{13}$$

This works by counting how many times a word appears in the entire document, or in our case the entire corpus, and weights the words inversely proportionally to this. This reduces the effect of words that appear throughout the corpus, typically words that carry little meaning. This means that the classifier is able to focus on words that are specific to the token and therefore the label.

4.1.3 Word2Vec

The previous SVM model used a simple count vectoriser, referred to as the bag of words (BOW), and TF-IDF transformer to vectorise and normalise the training samples of text. Whilst this method is simple it's simplicity has associated advantages and disadvantages. As the encoding for each word is a value between 1 and 0 there is no other few computations that can be done other than equality testing. This can severely limit the understanding and robustness of the model particularly to unseen data. From the example given in the previous section 15 you can see that the word "heart" would be given the same weighting for the second and third sentences. The TF-IDF encoding is only able to say that the word heart is present in both sentences. This may not be an issue for certain applications, where the database is sufficiently large and if the speed of training is an issue. However it's simplicity could result in a lower than possible accuracy for our classifier.

One of the main shortcomings of bag of words is that it does not encode the context or meaning of the word. It only looks at each word independently of the context of the sentence or paragraph and encodes it's frequency. For some applications this can result in a sub-optimal categorisation, such as subtler context driven sentiments such as social context. For most applications of classifying medical data with LeDeR it can be an issue due to the pen portraits. These contain often complex social indicators to the level of care received and as such a classifier must be developed that can identify these. This limitation is highlighted in Hughes et al.'s journal, "Medical text classification using Convolutional Neural Networks"[22]. Hughes suggests that sentences such as "the patient lives with her mother, who is not able to leave her home" are poorly represented with BOW . The sentence refers to the patient's mother a dictionary would not infer a relationship from this just two separate counts of "patient" and "mother". Secondly the sentence gives information about the patient's social situation without using language directly related to social situation. This would be easy for the human coder to recognise but would be near impossible for the dictionary method.

The bag of words model is also trained from only the training data. This means that the vocabulary of the classification is the same as the vocabulary of the training data. If a word not in the training data occurs for in a new instance the model would be unable to classify it. This could be an issue for LeDeR as a combination of a relatively small training corpus and wide range of language means that the vocabulary often does not cover language used in new instances.

Instead of using the simple BOW model we will apply and compare the results of a more complex model to test if a different vectorisation improves accuracy. The method we will test is word embeddings. The particular implementations are word2vec and doc2vec, this is a deep learning technique using a two layered neural network. Word embeddings work by instead of taking a one-hot encoding they take a vector of a set dimension, often in the order of 100's, and distribute a representation of a word across these dimensions. Each element in the vector contributes a different aspect to the meaning of a specific word. This enables more complex representations of words. The weightings are learnt via the the two layered neural network.

Word embeddings work by training on a very large corpus to learn a vector representation of words. In our case we will use the google news database, a model trained on 100 million words and phrases. Instead of each word corresponding to a word count, each word will be represented by a vector, the pretrained google news model uses a vector of dimension 300. This means that with a sufficiently large and diverse corpus the vector will represent the dominant or general meaning of the word. There are many ways to learn this vector representation of individual words.

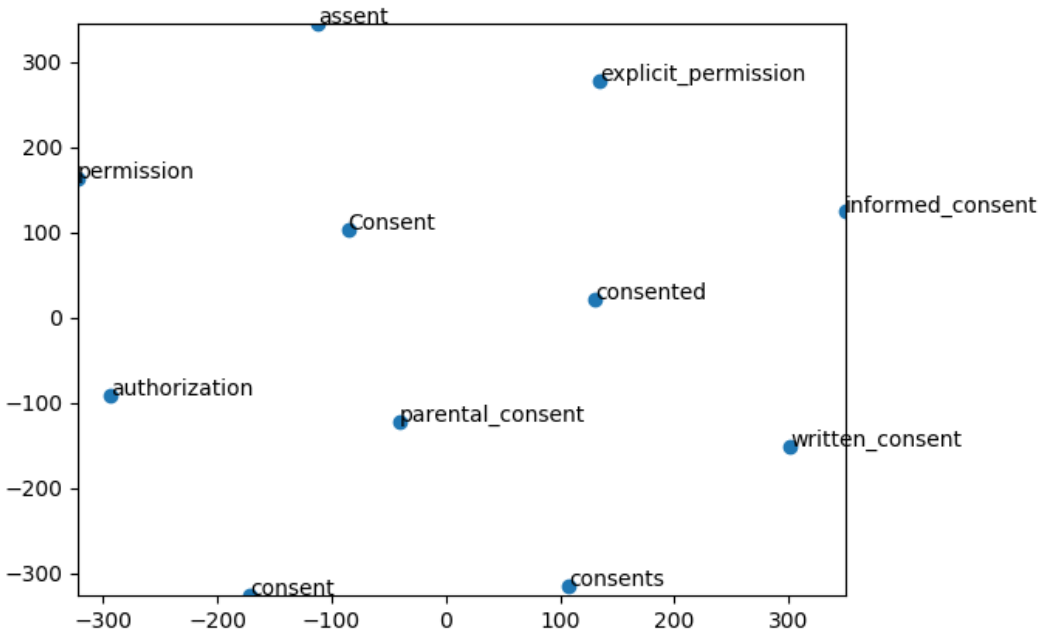
The word2vec algorithm is a combination of two techniques, continous bag of word method (CBOW) and a skip gram model. The CBOW works by learning and then predicting the proba-

bility of a word given the larger context of the sentence or document. The Skip gram method does the opsite of the CBOW. It uses the word to predict the context. The combination of these two models gives an algorithm that understands the context of the word at a more detailed level than a one-hot dictionary encoding and as such can represent words numerically with probabilities and vectors.

As the model understands similarity it is able to project from one word into other dimensions . Where previously "good" and "great" would have zero relation. This is not beneficial. Word embeddings are able to perform basic algebra on the learned vector representation to calculate similarity. This means that size of the input matrix can be reduced and the efficiency, and potentially accuracy can be improved.

For example in figure 16 the T-SNE dimension reduced representation of word similarity from the word2vec model is shown. Here we look at similar words to "care". .

Figure 16: T-SNE dimension reduced representation of word similarity for 'Consent". The word2vec model is using using the google-news-vector pre-built model.



For our data we will look at applying the word embeddings before the training and classification stage by applying both word level embeddings with word2vec and sentence level embeddings with doc2vec. Word2vec will generate a vector representation for each word input where doc2vec will generate a vector representation for each document, or in our case token, input. The different vectorisations can be compared for accuracy and efficiency.

4.2 Results

The performance of the models were assessed by testing on a sub-set of the original labelled data, referred to as the test set. As this was as this a domain specific supervised learning task, we are limited to data that has been labelled with the same labels as the classifier to test the accuracy. The same split was used for every method, the training set consisted of $47,498$ tokens whilst the test data-set was $20,357$ tokens long.

To compare performance between the two models we will look at three measures; precision, accuracy and the weighted F1-score. Precision is the model's ability to not label a negative sample as positive, a measure of the models robustness to false positives. This is important for the application as incorrectly labelling a token has a much greater impact than a false negative, it is more desirable for no new information to be added to the LeDeR database than incorrect information.

Precision is calculated by:

$$\frac{t_p}{(t_p + f_p)} \tag{14}$$

Where t_p is the number of true positives and f_p is the number of false positives.
Recall is the ability of the model to find positive samples, it is the models robustness to false negatives and is calculated by:

$$\frac{t_p}{(t_p + f_n)} \tag{15}$$

Where f_n is the number of false negative samples

The final measure is the weighted f1 score, this is a combination of both the precision and accuracy and gives an intuitive score for the overall performance of the model. The relative combination to the average for each score is equal. It is a measure of the incorrectly classified cases compared to the accuracy which is simply a measure of the correctly classified cases.

The formula for the F1 score is:

$$F1 = \frac{(P \cdot R)}{(P + R)} \tag{16}$$

Where P is the precision and R is the recall, previously defined.

The performance of the two models is summarised in the following table for all three measures. The table shows three different average weightings for the three measures of both models performance. The averages are calculated across all labels for each model for the same test set.

	Precision		**Recall**		**F1-Score**	
	TF-IFD	**Word2Vec**	**TF-IFD**	**Word2Vec**	**TF-IFD**	**Word2Vec**
Accuracy					0.64	0.61
Macro Avg	0.64	0.62	0.63	0.60	0.63	0.60
Weighted Avg	0.64	0.61	0.64	0.61	0.63	0.60

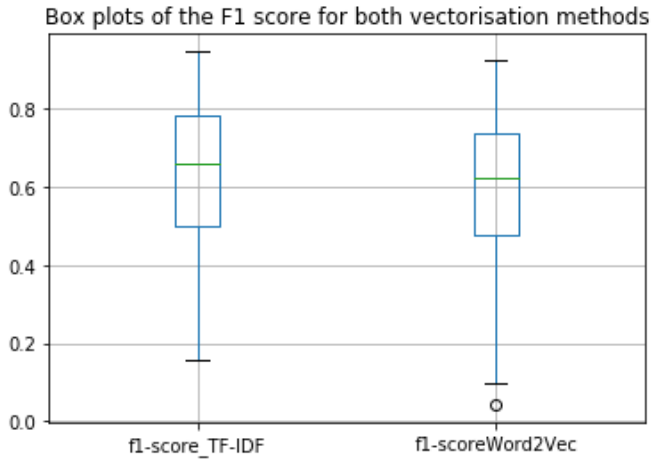
Table 1: Macro and weighted average for the precision, recall and F1-score for both models. The accuracy is also shown.

Table 1 shows both models having an F1 accuracy score of over 0.60%, the TF-IDF model has an accuracy of 64.07% with the word2vec's accuracy being 60.5% This means that the classification models achieve the first and second goals of the project as the desired accuracy is achieved over

the target number of labels. This means that as the model stands we have reason to believe that if implemented correctly the model has sufficient accuracy to improve the labelling time of coders.

The precision and recall scores are higher for TF-IDF than Word2vec for every single measure indicating that the TF-IDF would be the most suitable vectorisation technique from current development based on average accuracy alone. Figure 17 shows two box plots for the F1 score for both vectorisation techniques. Highlighting that the TF-IDF has a higher average accuracy for all labels. . Not only does it show that the overall accuracy is higher for the TF-IDF vectorisation

Figure 17: Box plots for the F1 score over all labels for both vectorisation techniques



technique but the spread of F1 scores over the labels is also tighter. The standard deviation is 0.1904 for TF-IDF compared to 0.1953 for Word2Vec. Whilst the overall accuracy for TF-IDF is higher than word2vec there are some key classes where word2vec significantly outperforms TF-IDF.

Table 2 highlights the largest label performance discrepancies between the two vetorisation techniques.

	Precision		**Recall**		**F1-Score**		**Support**	
	TF-IFD	**Word2Vec**	**TF-IFD**	**Word2Vec**	**TF-IFD**	Word2Vec	**TF-IDF**	**Word2Vec**
MCA Comments[1]	0.75	0.49	0.80	0.49	0.78	0.49	844	232
Care plan EOL[2]	0.68	0.30	0.78	0.23	0.72	0.26	468	301
Gd coordination[3]	0.46	0.90	0.34	0.95	0.39	0.92	346	193
Gd practise Primary	0.42	0.68	0.35	0.59	0.38	0.63	246	194
Constipation	0.90	0.49	0.80	0.38	0.85	0.43	267	198
Support Primary[4]	0.50	0.92	0.64	0.92	0.56	0.92	205	215
Dementia	0.88	0.64	0.91	0.61	0.90	0.63	224	174
Family brought in	0.52	0.90	0.41	0.92	0.46	0.91	182	331

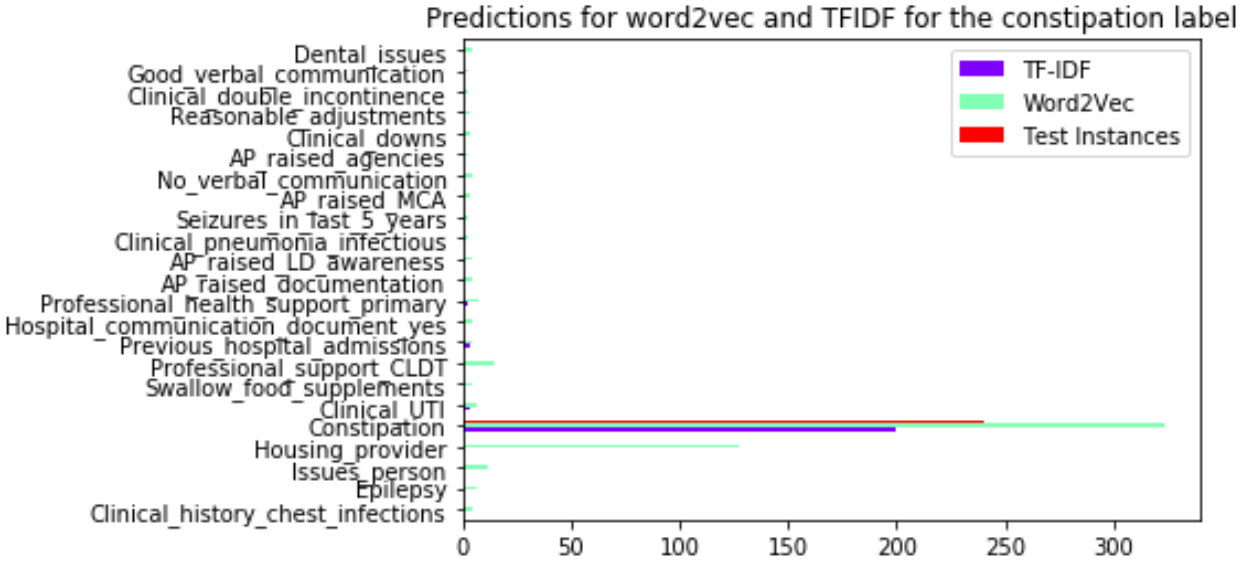
[1] Mental Capacity Act comments
[2] Care plan End Of Life.
[3] Good care coordination.
[4] good support primary.

Table 2: Notable results differences per label for SVMs with a TF-IDF vs Word2vec vectoriser. Results are shown for precision, recall and F1-score for both models for each label.

One particular area that the TF-IDF method excelled in and the Word2Vec failed was labels with specific associated medical language. For example the "Constipation" label performed well

with TF-IDF achieving an F1 score of 0.85 compared to Word2Vec's F1 score of 0.43. This could be as the label is heavily associated with specific language unique to that label, of the 825 training and test texts 687 tokens included the words "Constipation" or "constipated". This equates to 83% of all samples. An appropriate vectorisation technique should be able to weight this specific language in such a way that it can easily be distinguished from irrelevant medical language for classifier. Word2Vec failing to do this resulted in the large discrepancy in accuracy. Figure 18 shows the distribution of label assignments for the constipation code for both vectorisation techniques.

Figure 18: Distribution of Word2Vec and TF-IDF label assignments for the "Constipation" label. Including Actual number of instances within the test set.

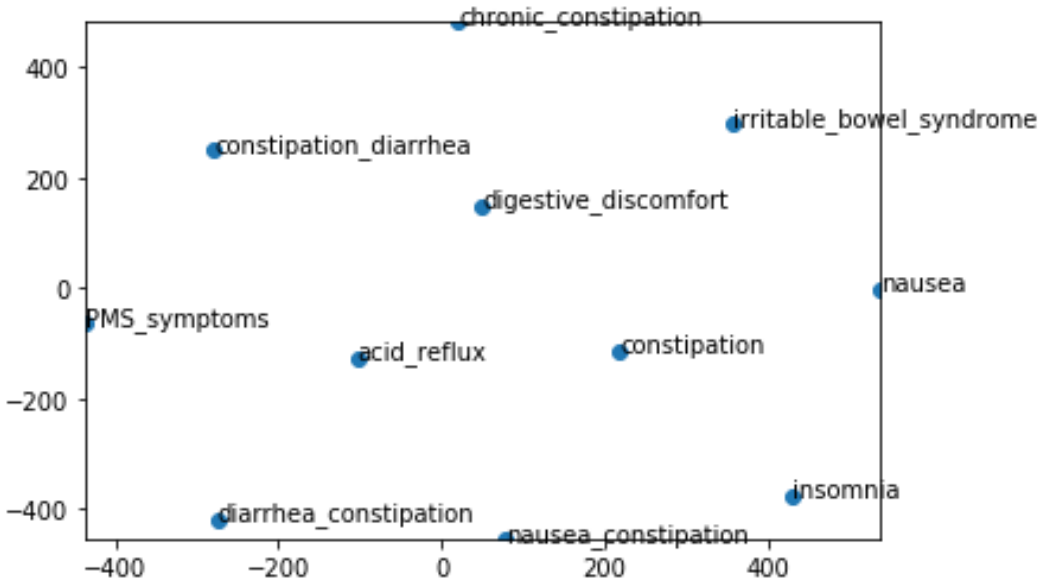


.

The word2vec vectorisation technique struggles with these hard boundary rules, where a single word has a large effect on the classification. You can see from the graph that it has not only predicted a small number of instances over a large number of labels incorrectly but it has also predicted far too many instances for the constipation label. It has failed to vectorise the words in such a way that the hard decision boundary can be established by the SVM. The TF-IDF method does not fail for this label as it represents the word occurring as a single weight in a larger matrix. This means that the SVM can have a definite input, it can easily detect the medically specific language required for the hard boundary hence the better performance. The word2vec representation is defined less strictly; it instead represents the word "Constipation" as a vector, similar to a large number of non-specific or specific but irrelevant vocabulary . In figure 19 we can see the similarity map of the word "constipation" for the word2vec model used. The model being the google-news-vector pre-trained word2vec vectoriser [5]. . Whilst this gives some understanding of the general context and meaning of the word the word2vec model finds the most similar word to be nausea, this may be useful in non-medical applications however this is unhelpful for medical classification. This can be seen from the wide distribution of estimations, the word2vec vectorisation results in many of the "Constipation" labels being classified as other medical conditions such "Epilepsy". This could be because there are limited medical documents in the training corpus and as such there will be few uses of medical language such as "Constipation".

[5]https://github.com/mmihaltz/word2vec-GoogleNews-vectors

Figure 19: Vector similarity for the word "constipation" from the google news data pretrained word2vec model



This limits the ability of the classifier to classify these specific cases as the vectorisation does not differentiate sufficiently between specific medical language.

In some extreme cases the language used in the pen portrait does not occur in the word2vec vocabulary, this means that the words would be unvectorised and would not be inputted into the classifier. The coverage of the word2vec model is graphed in figure 20, that being the percentage of the training and test vocabulary that is vectorised by the word2vec model, and the accuracy of predictions for the the label. We also plot a scatter graph with the size of the plots representing the number of samples in the set in figure 21. Both graphs show that there is a general trend that the greater the vocab coverage the more accurate the model. Whilst the the graphs show us the coverage of the word2vec transform they do not show it's understanding. The constipation example highlights this key difference.

There are however cases where word2vec outperforms TF-IDF. This can be seen in labels such as "good care coordination", "support primary" and "family brought in". These labels do not necessarily have unifying specific language. For example if we examine the "good care coordination" label more closely, of the 1096 examples in the training and test data sets only 41 contain the word coordination. A much smaller 3.7% compared to the "Constipation" label's 83%. As there is no single word to classify the label by the models understanding of the general meaning of the language is important. This results in word2vec having almost double the accuracy of TF-IDF for the good care practise label. Here we compare the word similarity map for coordination.

From the graph it is clear to see that the majority of words found to be similar are also relevant to the label. Similar input vectors require similar labelling outputs making the classification easier and more accurate.

Overall the results have highlighted that either model would give the desired performance for a classification tool used by the coders however there are also some clear shortcomings. The TF-IDF performs generally robustly with the word2vec model having two main shortcomings. Identifying medically specific language and the understanding of out vocabulary language.

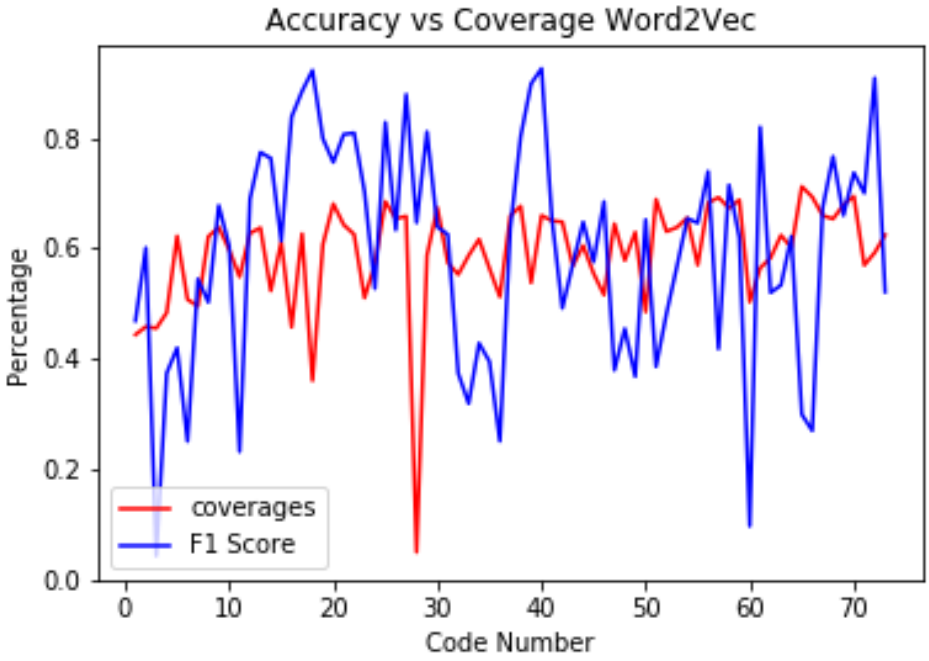


Figure 20: Accuracy vs Coverage for all Labels

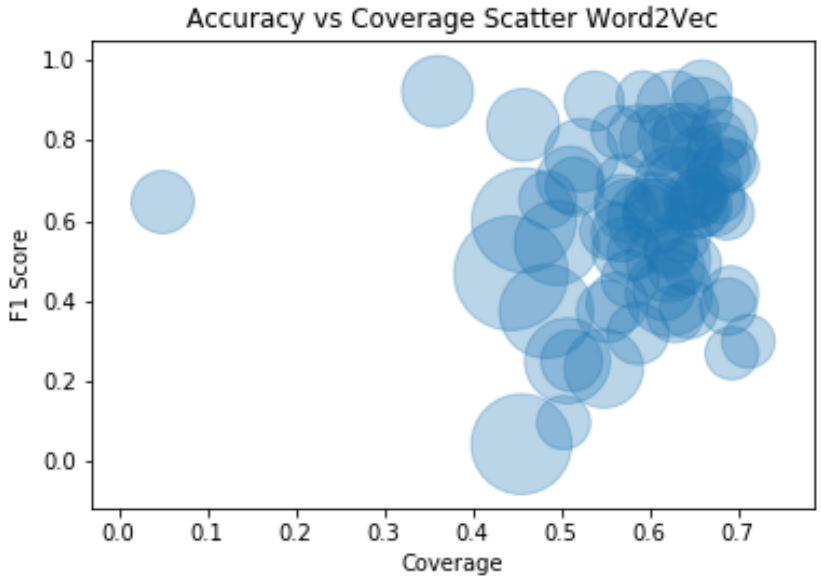


Figure 21: Accuracy vs Coverage Scatter Graph, Accuracy in the form of the weighted F1-score plotted on the Y-axis with vocabulary coverage of the test data-set by word2vec on the x-axis. The size of each point represents the size of the label corpus.

5 User Interface

With an effective classifier established it must be implemented in such a way to achieve the goals of the project, to do this the interface must be fast and easy to use. The ease and speed of the implementation could have as great an effect on the utility of the program than the accuracy of

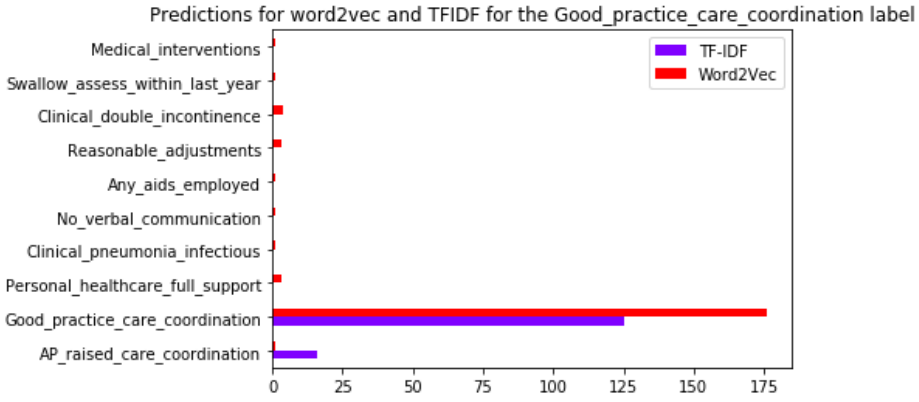


Figure 22: Distribution of Word2Vec and TF-IDF label assignments for the "Coordination" label.

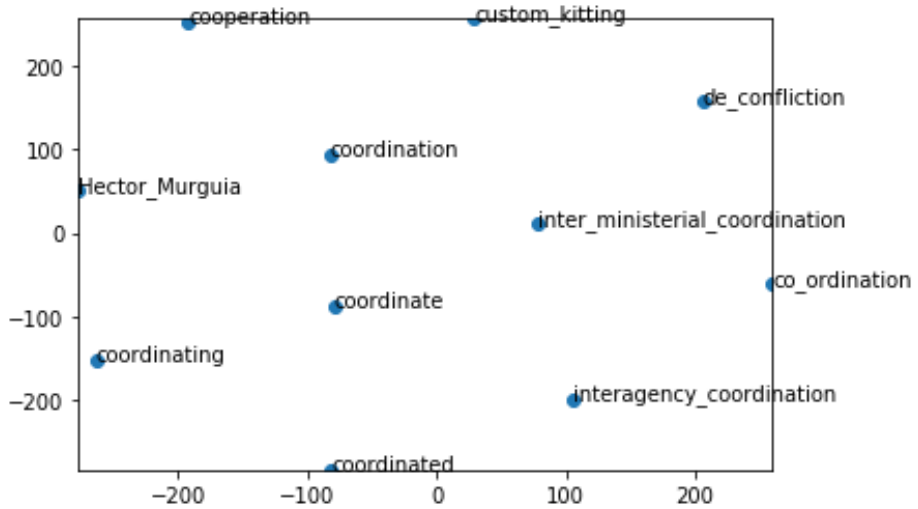


Figure 23: T-SNE dimension reduced representation of word similarity for 'Coordination". The-word2vec model is using using the google-news-vector pre-built model.

the classifier. A basic but effective Graphical User Interface (GUI) is proposed to quickly display the information required by the coders. The GUI must enable the coders to make a reasoned and justified choice, the coder must be able to make a decision on the code using both the knowledge from the classifier and their own prior knowledge. Simply displaying labels would not indicate the reasoning of the model and would impair the coders decision.

There are three aims for the design of the GUI:

1. To improve the speed of which coders can code a document.
2. To improve the accuracy of which coders can code a document.
3. To quantify the models confidence so that coder can evaluate and make an informed code selection.

5.1 Methods

The first design decision that had to be made is the platform to build the interface on. There are many options from web based applications to command line interactions and stand alone apps.

The GUI will first be built as a local app to maintain the security of the system, this will use data only from the already established LeDeR secure drive and also run on the local LeDeR drive eliminating the need for any external network access, a potential weak point for security. Developing tools outside of the existing infrastructure would implausible as any expansion to the existing network could introduce security risks. A locally run application should also improve the speed of the program. No data will have to be retrieved from external servers potentially increasing running time.

The program will be built using a python module TKinter [6]. The Tkinter module ("Tk interface") is the standard Python interface to the Tk GUI toolkit. This is a computationally light GUI interface, it also benefits from running on python, the language used to build the classifier, reducing the potentially complex interfacing between different languages. As TKinter is a simple GUI development tool it is quick to develop and therefore works well as a proof of concept.

5.1.1 Input

The reports arrive to LeDeR as a single word file. Within this word file is the pen portrait in a box object, this is then exported to the LeDeR database. The coder will then highlight the text in the program and assign codes. As there are several stages to the coding process there are many points at which the text can be labelled by the program. The earliest of these is by inputting the original word file straight to the classifier. Alternatively it could be inputted when the pen portrait is added to the LeDeR database. Finally the text can be inputted when the coder is coding the text within their own application. It must also be decided how the text is tokenised. This is the process of splitting up the large chunk of pen portrait text into smaller classifiable "tokens" so that sentence level sentiment analysis can be performed. For the data previously collected the text was tokenised by the coders. This was irregular so that token length varied as seen in the data section, 5,6.

The first option for handling input to the program would be to pass the word document straight into a tokeniser and classifying then the tokenised document. This could then be displayed to the coders with the labels and confidences of the model. This would be the most automated method, the coders would not have to learn how to use a new piece of software as the classifications would be displayed automatically, instead they would be able to open the pre-labelled document. Full automation does also have it's disadvantages. It would only be able to pass the document through it's own tokeniser, the quality of the program would be heavily dependant on the quality of the tokeniser. It would also allow minimum interaction between the coder and the classifications, the classifications would just be displayed and either be accepted or not. This would be beneficial for an unskilled coder however the program is intended to be used with the current, skilled workers. A greater level of interaction would prove advantageous. Similar issues would apply if the text was inputted when the pen portrait were transferred from the word file to the LeDeR database.

The final and most basic option would be to have a stand alone program running alongside the coding process. This could work by coders simply selecting text and inputting it into the classifier. This would be the most basic as no other interfacing would be required to retrieve the text data. The would also be practical as currently each coder has multiple monitors on their

[6] https://docs.python.org/3/library/tkinter.html

personal computers, the two program's could be run side by side with relative ease. This would also be a fairly versatile option. The coder could decide which text was inputted into the program and select as they wanted. This means that it could be a faster process, the easily classifiable codes could be identified quickly by the coders and then all sentences where they were unsure could be inputted into the program. They would also have the option of putting the entire text into the code and using the built in tokeniser. This means that the method would not be reliant on a tokeniser and could be used as the coder desired to best fit their work-flow.

Tokenisation in NLP refers to the act of splitting up text into smaller chunks, for sentences this would be splitting into individual words and for paragraphs splitting it into sentences. There are many approaches that can be taken most are either word level tokenisation or sentence level. This means that designing a tokeniser that replicated tokenisation done by the coders would be incredibly computationally complex.
The coders do not tokenise based on sentences but sentiment. This means each token will have one or more consistent sentiment resulting in a variation of token length. This means that tokenisation is a current limitation and as such the input method must consider this.

As the tokeniser could be the largest limitation the first implementation will be the coders inputting the text when coding. This program would allow the coders to select to use the automatic tokenisation or tokenise the sentences themselves. This should result in a robust classification tool.

Figure 24 shows a sample of the input screen.

Figure 24: A screenshot of the input interface for the program, the input sample is from the report featured in the appendix

```
Enter Text to Classify>>> Carrie had Downs syndrome and a moderate learning disability. She was born and lived in Wolverhampton all her
life. Carrie had an older sister, Sandra (Sandy) and throughout her life she and Sandy were very close, perhaps because their mother died
while they were very young (Carrie was three). Following Carrie's father having a stroke at the age of 60, Carrie also became a carer.
Carrie was able to run errands; supporting her dad to maintain the house and go to the local shops with a list. Sandy visited daily to cook
and complete other tasks. Six months before their dad died he needed to go to nursing care and Carrie, then aged 25, went to live with
Sandy and her family where she remained for about 10 years, until she moved to independent living in 2012. In her mid 30s it was noticed
that Carrie had some difficulty seeing smaller items, and visited an optician in who had a daughter with learning difficulties. Tests
showed that Carrie had always had defective vision in one eye, and she was delighted when she received glasses to be able to read bus
numbers at a distance. Carrie would also have benefited from wearing a hearing aid however she didn't want to wear one as she associated
this with being old and she always said she never wanted to get old
```

5.1.2 Output

The program must be fast to interpret and justifiable, the output needs to display the classification in an easy to read way. To achieve this the classification and the text are displayed in the same place. This means the coder does not have to search for the classification wasting valuable time. We also give the text for classification a number so that it can be referred back to if required. We also must make it justifiable, this means that the coder understands the decision reached by the classifier and assesses this in relation to their knowledge. To do this the confidence in the prediction will be displayed. As a confidence level we will use the hinge loss function, whilst this is not an exact probability it is a measure of how close an example is to the decision margin so can be used as a rudimentary confidence measure. This confidence measure will be displayed alongside the classification.

The final decision is to how the output assigns multiple labels. As this is a multi-label task it is important that if multiple labels can be assigned to the token. This initially was just by displaying the top three codes. This did not handle tokens where no code was applicable meaning coders would waste time selecting between multiple labels each with very low probabilities. Therefore we

introduce a minimum threshold so that only codes over a certain threshold are displayed. Sample output is shown in figure 25 below.

Figure 25: Sample output of the program, the program has been run with auto-tokenise with the same input as shown in the input sample 24. The full pen portrait can be found in the appendix

```
 Classified Text

 '
Text ID: 0
Text: Carrie had Downs syndrome and a moderate learning disability.
Label: ['Mobility_some_support']
Confidence: [0.9690134163613596]

Text ID: 1
Text: She was born and lived in Wolverhampton all her life.
Label: []
Confidence: []

Text ID: 2
Text: Carrie had an older sister, Sandra (Sandy) and throughout her life she and Sandy were very close, perhaps because their mother died
while they were very young (Carrie was three).
Label: ['Fam_rel_seemed_good']
Confidence: [0.5903439786887931]

Text ID: 3
Text: Following Carrie's father having a stroke at the age of 60, Carrie also became a carer.
Label: []
Confidence: []

Text ID: 4
Text: Carrie was able to run errands; supporting her dad to maintain the house and go to the local shops with a list.
Label: ['Personal_healthcare_some_support']
Confidence: [0.5654848110573371]

Text ID: 5
Text: Sandy visited daily to cook and complete other tasks.
Label: []
Confidence: []

Text ID: 6
```

6 Discussion

6.1 Validity of Results

In 75 of the LeDeR pre-determined labels results were found for both a TF-IDF and Word2vec word vectoriser. To understand the results fully and evaluate how well the aims of the project have been met it is essential to interrogate the validity of these results.

The accuracy is comparable between vectorisers because the methods for testing the vectorisers were identical, the training test split was the same for both vectoriser methods and as such the accuracy scores are generated from identical data. The accuracy of both classifiers on average is similar, this is reconfirmed in prior studies on comparisons between TF-IDF and word2vec. Zhu et al. test an SVM, amongst other classifications, in the domain of traditional Chinese medicine to classify text [23]. They find Word2vec and TF-IDF vectorisation methods to achieve similar results. TF-IDF achieved a precision score of 0.823 with word2vec scoring 0.811 for an SVM classifier. This validates our results as the precision for both vectorisation methods precision's are close in value, this directly compares to the results found for the same vectorisers when applied to the LeDeR data: The precision scores for both methods are concurrent. Zhu et al. find accuracy scores for TF-IDF and word2vec respectively to be 0.676 and 0.792, considering the change of scope, 2 classes compared to the 75 used for LeDeR, the decrease in accuracy to 0.64 and 0.61 for TF-IDF and word2vec is minimal, validating that the accuracy of the classifier is on par with similar studies.

Both methods met target accuracy for predictions as outlined in the goals of the project, to fairly evaluate the success of the project it is crucial to consider the validity of the target accuracy. A classifier alone cannot accomplish the goals of reducing time to code the labels, it must be used in conjunction with a user interface. The current target accuracy of the classifier is based on a study into content analysis using automatic text processing technology[5], therefore the domains must be compared so that the use of the target accuracy can be justified. The minimum accuracy required to reduce the time of coding for a similar user interface which was found to be 50%. Whilst the tasks are similar the classifier domains differ. Gweon et al.[5] use a coding scheme established by Witter et al.[24] a study in automating technological advice for consumers. In the study by Gweon et al. the coders would read a 6 page coding manual and then be separated into a control group and a test group to compare a semi-automated code and correct model versus a fully manual coding method. The coders are then given queries, in the form of on-screen text, to code one query at a time. The text would be labelled either through recalling a memorised set response or by accepting or correcting a automatically generated label. The method tested by Gweon et al differs from the LeDeR application as the coders were not experienced professionals (as the LeDeR coders are) but students who had undergone basic training. This could potentially result in the semi-automated model having a greater effect on reducing time for the inexperienced coders. The coders in the Gweon et Al's study would require extra prompting as they are less familiar with the coding scheme, this must be considered when evaluating the accuracy of the classifier.

The scope of the classification also differs, Gweon et al only make a classification for ten codes. This is far less complex than the seventy-five codes used in the LeDeR predictor. The classification task is also less complex as it is multi-class not multi-label. One query is shown at a time and the coders are required to only assign one label to this query. Once one label has been found no other label needs to be considered, this is not the case with LeDeR. The format of the coding platform also has an effect. Gweon et al's coding format presented a series of individual tokenised sentences to be coded compared to an entire pen portrait that needs to be both tokenised and categorised,

the LeDeR data adds another degree of complexity in this respect.

As the LeDeR domain is potentially less sensitive to the effects of introducing a semi-automated model and also requires a more complex classification domain an increase in target accuracy is justified. In this case 60% is selected however this is an estimation. Therefore the validity of the accuracy of the classifier hinges on the limited validity of the target accuracy. By increasing the target accuracy and in turn the accuracy of the classifier the validity of results would be improved. Despite this, increasing the accuracy is non-trivial and 60% is a reasonable target. To be fully valid any reduction in time must be tested and quantified in the exact setting it is intended to be used. The most severe limitation of the project is that the required accuracy is only an estimation.

6.2 Further Work

With the time restrictions of the project there are several key areas in which further work can take place. Despite external sources suggesting that the accuracy achieved by the classifier should be sufficient to reduce the time taken to label the text [5], validation will be needed for the specific LeDeR setting. This initially means carrying out the experiments outlined in the testing process section 9.2. It is imperative that all future work is based on these experimental results. With the experimental results in mind there are two areas in which the project can be developed. The accuracy of the classifier and the quality of the user interface. Without experimental data the current limiting factor is unidentifiable thus improvements are proposed to both sections.

6.2.1 Classification

The current classifier meets the target accuracy outlined previously, this can still be improved. The two vectorisation methods, TF-IDF and word2vec, achieve a similar accuracy but in differing classes. The TF-IDF performs well for classifiers with label-specific medical language whereas the Word2vec method performs better for vague language sentiment. Improving accuracy of the TF-IDF method is challenging, the vectorisation method is an untunable vectoriser. Improving it's accuracy can be done by either increasing the training set, which is unfeasible, or by improving or changing the classifier. Several improvements have been outlined in the prior studies section 3 including developing a neural network. This could allow for harder margins and potentially increase accuracy but will not necessarily represent the labels with less specific sentiment more accurately. Therefore, the logical progression would be to improve the performance of the word2vec model.

The apparent weaknesses of the word2vec is due to its lack of understanding of specific medical language. By improving this understanding it would consequently increase the performance of the model. The current version of word2vec is trained using the google news data-set[7], the limited medical language of the corpus has a detrimental effect on the results of the model. The word2vec vectoriser can easily be adapted to include a larger and more targeted corpus. The strength of word2vec compared to TF-IDF is that building the vectoriser beyond the corpus of the training data can increase its understanding when predicting over the training data. For example the current corpus of news data can be changed for pre-trained medically specific models, one such model is Biowordvec [19]. This uses the MeSH data-set[8], a large collection of unlabelled medical texts of roughly 4.5b words. This is a large increase in the size of the corpus used and includes many medical terms not present in the current vectoriser increasing its coverage. Not only does

[7]https://github.com/mmihaltz/word2vec-GoogleNews-vectors

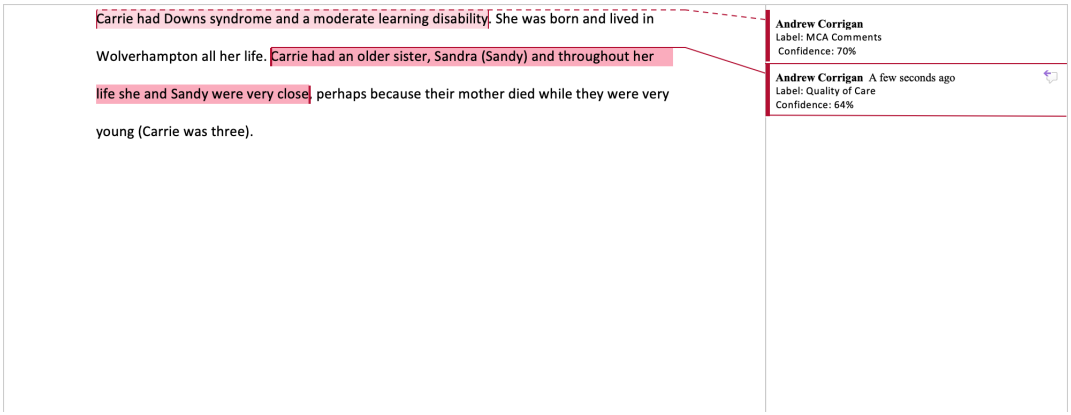
[8]https://www.ncbi.nlm.nih.gov/mesh/

the improved vectoriastion method increase the corpus of the vectoriser but also considers "sub-word information". This breaks up complex medical words and compares individual components for a better representation of medical word similarity. For example the words "paediatrics" and "paediatrician", the sub-word "ped" is common between both and therefore will be vectorised as similar words. This is proven to improve performance for medical classification tasks[19]. Pairs of medical words were presented to a number of medical professionals and the similarity of the pairs was recorded and averaged. The world similarities were also computed for the word2vec model trained on news data and also computed for the biowordvec model. The biowordvec model was found to significantly outperform the basic model for word and sentence similarity tasks. This improved vectorisation technique could easily be applied to the LeDeR data-set and could drastically improve accuracy for medically specific labels whilst maintaining the accuracy of the subtler labels.

6.2.2 User Interface

As it stands the user interface (UI) is basic and acts as a proof of concept. It therefore has a large scope for improvement. Before any improvement to the user interface is made it is vital that the effectiveness and limitations of the current user interface are understood: The user interface must be tested in the environment it is designed to be used in. Due to limitations on access to the LeDeR office and personnel as a result of the current situation with COVID-19, these experiments could not take place. A detailed plan of exactly how the experiments would be carried out is included in the appendix 9.2. In leiu of these results potential improvements to the user interface are suggested. The current UI runs as a separate program on each coders machine, they have to proactively input text from the current coding program into the user interface. This is a new addition to their workflow and as such requires a conscious effort to use. It also outputs the predictions in a different format to the original word file, it is a potentially time consuming process to find and relate the classified output to the original text. A reasonable extension would be to improve the speed of use. This could be improved by moving the output to a web app, this web app could output the classified text in the same format as the original word document, as untokensied text, but highlight sections of interest. This would reduce time as coders would quickly be able to identify the key sections of the pen portrait as identified by the classifier. A mock of what this could look like has been created using a basic word interface in figure 26. Here the code is

Figure 26: A mock of proposed improvements to the GUI.



highlighted using the word annotation tool an alternative to the web based app. This uses existing

tools so can be quickly implemented. The move to a centralised app could also benefit future iterations of the classifier. Updates to the classifier such as improved weighting or new labels could be pushed to every machine simultaneously rather than coders being responsible for keeping the classifier up to date.

It is worth noting that currently LeDeR uses NVIVO, this is a closed source program and as such no API integration is available. Directly integrating a highlighting or semi-automated labelling tool to NVIVO would be unfeasible. The only way to avoid running multiple applications would be to create a stand alone labelling program however the work required would outweigh the benefits of the time reduction from a semi-automated method.

7 Conclusion

The primary aim of the project was to increase the speed, accuracy and efficiency of the LeDeR programmes labelling process. This involved creating and testing a machine learning classifier and then implementing this classifier into a semi-automated coding process that is usable by the current staff. With this in mind the aims of the classifier were established, it was decided that to achieve the design goals of the end product the classifier would need to achieve a minimum of 60% accuracy over 60% of all instances. Once this classifier had been developed and optimised it could then be implemented into a usable program.

The first steps were to evaluate the scope of the data that needed classifying, so that existing methods and previous results could be compared to establish which method would be most effective for the LeDeR data. This was done through various metrics. Common approaches for similarly structured data that achieved desirable results used an SVM as a based classifier. This was therefore selected as the first model.

The effects of two different vectorisers are then compared, TF-IDF and word2vec. The overall accuracy as well as the accuracy per label are compared to find the strengths and weaknesses of each vectorisation technique. The TF-IDF had a stronger performance for medically specific labels with Word2vec outperforming TF-IDF on labels with vaguer unifying sentiment. Whilst both models achieved the goals required for implementation the overall accuracy was found to be higher for TF-IDF so it was selected as the classifier for use in the UI. The differences in performance do suggest that for future developments that the word2vec model could be adapted to give greater accuracy than both current classifiers with existing developments from the Biowordvec model [19].

The classifiers developed were tested on a subset of the initial data. This means that in the isolated sample of LeDeR data previously collected the methods were shown to be successful. For the current scope of the project, developing a classifier for implementation for just the LeDeR programme, the results are valid. Therefore if the classifiers were to be applied to a wider medical context the performance would need to be reassessed, similarly the validity would also be affected if trends in the LeDeR data changed. For example if the language used changed over time or new labels were introduced to incorporate modern medical trends. One example of a medical trend is the current COVID-19 pandemic, LeDeR has recently seen a huge influx of death reports due to the pandemic and therefore the utility of speeding up the coding process is as important as ever. The program could adapted to learn new labels such as one which describes the effects of COVID-19 provided that there is sufficient training data. Even without pandemics it would be sensible to periodically re-train the classifiers with data recently classified. This could easily be implemented so that the accuracy could improve as the training set increases.

Once the classifiers had been developed and the limitations of the classifiers understood, the

classifiers were implemented as a simple graphical user interface. This allows coders to input tokenised text or have the program tokenise the text automatically. The text would then be classified and the classification, confidence levels and original text is outputted to a simple on screen display. The effectiveness of this is yet to be assessed as due to the current climate testing could not take place. In it's place a set of experiments have been outlined that will be able to provide results to show if the initial goals had been achieved. In lieu of these results improvements to the implementation of also been suggested, specifically developing a centralised web app and improved interface.

Overall the project has created multiple classifiers that have been proven to be valid and as far as currently testable the accuracy of these classifiers is great enough to meet the original targets of the project. It has shown through current natural language processing techniques a usable interface has been developed to aid coders with research into reducing an unjustifiable gap in life expectancy. With testing the application can be delivered to the LeDeR programme and used effectively with minimal extra training of the coders. Several points of improvement have also been highlighted and can be easily implemented to improve the effectiveness of the application. The robustness of the application mean that ongoing improvements can also be made easily be made. Resulting in a program that can not only be implemented for the short-term but would also be useful as a versatile long-term solution.

8 Mitigation Table

Event	Impact on Report	Actions taken to mitigate effect	Remaining Impact
Unable to access data for the last two weeks of the project to generate more results	The results section was lacking depth, important plots such as confusion matrixes were not plotted	Performance metrics are provided where possible, as many results were plotted as possible.	The results do not go into as much problem-specific detail as would have liked, e.g. training time impacts of the test train split
Unable to test implementation of the classifier due t o lockdown	The actual effectiveness of the program was not able to be assessed, so results are missing for the second section of the report	The focus of the report has been shifted to the classifier where originally the focus would have been on the implementation.	The breadth of the report has been reduced, originally the focus would have been on an original idea rather than implementing existing methods The amount of possible cross-d work has been reduced.

References

[1]NHS, "Statistics on care of people with learning disabilities", Accessed: 2019-11-24.

[2]J. Thornton, "People with learning disabilities have lower life expectancy and cancer screening rates", BMJ **364** (2019).

[3]J. Saldaña, *The coding manual for qualitative researchers* (Sage, 2015).

[4]"Multi-label classification with one-vs-rest strategy", Accessed: 2020-05-15.

[5]G. Gweon, C. P. Rosé, J. Wittwer, and M. Nueckles, "Supporting efficient and reliable content analysis using automatic text processing technology", in Human-computer interaction - interact 2005, edited by M. F. Costabile and F. Paternò (2005), pp. 1112–1115.

[6]*Data breaches: in the healthcare sector*, `https://www.cisecurity.org/blog/data-breaches-in-the-healthcare-sector/`, Accessed: 2020-05-20.

[7]*Nearly 200,000 patients impacted by pih health phishing attack*, `https://www.hipaajournal.com/nearly-200000-patients-impacted-by-pih-health-phishing-attack/`, Accessed: 2020-05-15.

[8]*Amca data breach*, `https://www.advisory.com/daily-briefing/2019/08/13/data-breach`, Accessed: 2020-05-15.

[9]*Scikit k-fold validation*, `https://scikit-learn.org/stable/modules/cross_validation.html`, Accessed: 2020-05-15.

[10]M. H. Stanfill, M. Williams, S. H. Fenton, R. A. Jenders, and W. R. Hersh, "A systematic literature review of automated clinical coding and classification systems", Journal of the American Medical Informatics Association **17**, 646–651 (2010).

[11]A. Eyecioglu and B. Keller, "Twitter paraphrase identification with simple overlap features and svms", in Proceedings of the 9th international workshop on semantic evaluation (semeval 2015) (2015), pp. 64–69.

[12]K.-J. Lee, Y.-S. Hwang, and H.-C. Rim, "Two-phase biomedical ne recognition based on svms", in Proceedings of the acl 2003 workshop on natural language processing in biomedicine-volume 13 (Association for Computational Linguistics, 2003), pp. 33–40.

[13]Y.-W. Chang, C.-J. Hsieh, K.-W. Chang, M. Ringgaard, and C.-J. Lin, "Training and testing low-degree polynomial data mappings via linear svm", Journal of Machine Learning Research **11**, 1471–1490 (2010).

[14]S. Nayak, R. Ramesh, and S. R. Shah, "A study of multilabel text classification and the effect of label hierarchy", in (2013).

[15]G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Džeroski, "An extensive experimental comparison of methods for multi-label learning", Pattern recognition **45**, 3084–3104 (2012).

[16]Y. Li, K. Bontcheva, and H. Cunningham, "Using uneven margins svm and perceptron for information extraction", in In proceedings of ninth conference on computational natural language learning (conll-2005 (2005).

[17]T. Baumel, J. Nassour-Kassis, M. Elhadad, and N. Elhadad, "Multi-label classification of patient notes a case study on ICD code assignment", CoRR **abs/1709.09587** (2017).

[18]L. Yao, C. Mao, and Y. Luo, "Clinical text classification with rule-based features and knowledge-guided convolutional neural networks", BMC medical informatics and decision making **19**, 71 (2019).

[19]yijia zhang, qingyu chen, zhihao yang, hongfei lin, and Z. Lu, "BioWordVec: Improving Biomedical Word Embeddings with Subword Information and MeSH Ontology", (2018).

[20]J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining", CoRR **abs/1901.08746** (2019).

[21]*In-depth: support vector machines*, `https://jakevdp.github.io/PythonDataScienceHandbook/05.07-support-vector-machines.html`, Accessed: 2020-05-15.

[22]M. Hughes, I. Li, S. Kotoulas, and T. Suzumura, "Medical text classification using convolutional neural networks", Stud Health Technol Inform **235**, 246–50 (2017).

[23]W. Zhu, W. Zhang, G.-Z. Li, C. He, and L. Zhang, "A study of damp-heat syndrome classification using word2vec and tf-idf", in 2016 ieee international conference on bioinformatics and biomedicine (bibm) (IEEE, 2016), pp. 1415–1420.

[24]J. Wittwer, M. Nuckles, and A. Renkl, "Can experts benefit from information about a layperson's knowledge for giving adaptive explanations?", in Proceedings of the annual meeting of the cognitive science society, Vol. 26, 26 (2004).

9 Appendix

9.1 Full Results

	Precision		Recall		F1	
Code	**TF-IDF**	**Word2vec**	**TF-IDF**	**Word2vec**	**TF-IDF**	**Wor**
MCA_comments	0.75	0.43	0.80	0.52	0.78	0.47
DNACPR_comments	0.76	0.66	0.82	0.55	0.79	0.60
Clinical_condition	0.51	0.28	0.74	0.02	0.60	0.04
Issues_agencies	0.48	0.31	0.70	0.47	0.57	0.37
Best_practice	0.24	0.46	0.28	0.39	0.26	0.42
Care_plan_EOL	0.68	0.30	0.78	0.21	0.72	0.25
Clinical_history_chest_infections	0.78	0.66	0.82	0.46	0.80	0.54
Fam_rel_seemed_good	0.56	0.49	0.60	0.51	0.58	0.50
Fam_contact_frequently	0.61	0.51	0.62	1.00	0.61	0.68
Epilepsy	0.80	0.65	0.79	0.57	0.80	0.60
Issues_person	0.53	0.21	0.54	0.26	0.53	0.23
Good_practice_care_provider	0.39	0.64	0.47	0.75	0.43	0.69
Vision_problems_yes	0.92	0.84	0.91	0.71	0.91	0.77
Clinical_cardiovascular_problem	0.84	0.77	0.78	0.75	0.81	0.76
AP_raised_care_coordination	0.39	0.55	0.36	0.69	0.37	0.61
Medical_histories	0.52	0.80	0.43	0.88	0.47	0.84
Good_practice_care_coordination	0.46	0.92	0.34	0.85	0.39	0.88
Weight	0.88	0.91	0.90	0.93	0.89	0.92
Swallowing_issues_dysphagia	0.70	0.76	0.68	0.84	0.69	0.80
Reasonable_adjustments_made_other	0.41	0.78	0.29	0.74	0.34	0.76
Communication_abilities	0.44	0.85	0.51	0.77	0.47	0.81
AP_raised_other	0.40	0.85	0.34	0.77	0.37	0.81
Clinical_pneumonia_aspiration	0.76	0.76	0.77	0.65	0.76	0.70
Behaviour_problems	0.70	0.45	0.68	0.64	0.69	0.53
Personal_healthcare_full_support	0.70	0.88	0.73	0.78	0.71	0.83
Mobility	0.41	0.61	0.44	0.66	0.42	0.63
Good_practice_primary	0.42	0.88	0.35	0.87	0.38	0.88
BMI	0.94	0.61	0.91	0.69	0.93	0.65
Nursing_CLDT_nurse	0.71	0.84	0.72	0.78	0.72	0.81
Housing_provider	0.85	0.64	0.92	0.64	0.89	0.64
Constipation	0.90	0.57	0.80	0.68	0.85	0.62
Clinical_UTI	0.83	0.45	0.81	0.32	0.82	0.37
Swallow_food_supplements	0.77	0.44	0.76	0.25	0.77	0.32
Professional_support_CLDT	0.62	0.37	0.71	0.50	0.66	0.43
Clinical_history_falls	0.73	0.43	0.76	0.37	0.74	0.39
Previous_hospital_admissions	0.53	0.37	0.54	0.19	0.54	0.25
Hospital_communication_document_yes	0.69	0.70	0.81	0.58	0.75	0.64
Good_practice_secondary	0.30	0.78	0.20	0.83	0.24	0.80
Professional_health_support_primary	0.50	0.90	0.64	0.89	0.56	0.90

Mobility_total_support	0.61	0.90	0.48	0.96	0.54	0.93
AP_raised_documentation	0.47	0.65	0.45	0.66	0.46	0.65
AP_raised_LD_awareness	0.48	0.42	0.55	0.59	0.51	0.49
Clinical_pneumonia_infectious	0.72	0.58	0.80	0.57	0.76	0.57
Nursing_LD_liaison_nurse	0.70	0.63	0.77	0.66	0.73	0.65
Immunised_against_flu_yes	0.88	0.66	0.95	0.51	0.92	0.58
Seizures_in_last_5_years	0.65	0.63	0.63	0.75	0.64	0.68
AP_raised_MCA	0.76	0.39	0.56	0.37	0.65	0.38
Last_health_check_within_1_year	0.74	0.47	0.77	0.44	0.75	0.45
Clinical_dementia	0.88	0.41	0.91	0.34	0.90	0.37
Medical_intervention_details	0.55	0.65	0.42	0.66	0.48	0.65
No_verbal_communication	0.74	0.40	0.62	0.37	0.67	0.39
AP_raised_agencies	0.23	0.51	0.09	0.46	0.13	0.48
Advocate_yes	0.71	0.57	0.56	0.56	0.62	0.57
Any_aids_employed	0.54	0.76	0.45	0.58	0.49	0.65
Clinical_kidney_problems	0.85	0.65	0.89	0.64	0.87	0.65
Mobility_some_support	0.46	0.69	0.49	0.80	0.48	0.74
Clinical_downs	0.92	0.48	0.94	0.37	0.93	0.42
Reasonable_adjustments	0.53	0.69	0.34	0.74	0.42	0.71
Clinical_double_incontinence	0.90	0.68	0.92	0.57	0.91	0.62
Medications	0.67	0.50	0.65	0.05	0.66	0.10
Swallow_assess_within_last_year	0.60	0.81	0.68	0.83	0.64	0.82
Pressure_sores	0.81	0.46	0.87	0.59	0.84	0.52
Epi_manage_descr	0.58	0.47	0.69	0.61	0.63	0.53
Skin_conditions	0.83	0.64	0.72	0.60	0.77	0.62
Good_verbal_communication	0.60	0.51	0.59	0.21	0.59	0.30
Dental_issues	0.90	0.36	0.87	0.22	0.89	0.27
Good_practice_EOL	0.44	0.69	0.32	0.66	0.38	0.67
Limited_verbal_communication	0.60	0.81	0.50	0.73	0.55	0.77
Personal_healthcare_some_support	0.75	0.68	0.67	0.64	0.70	0.66
Independently_mobile	0.77	0.72	0.69	0.76	0.73	0.74
Medical_interventions	0.65	0.74	0.39	0.67	0.49	0.70
Patients_family_brought_in	0.52	0.90	0.41	0.92	0.46	0.91
Poor_practice_care_coordination	0.56	0.87	0.17	0.37	0.26	0.52

9.2 Testing Process

Time restrictions and restrictions on access to the data resulted in the program being unable to be tested. Therefore as it is not possible to conduct any experiments a method for testing the effectiveness of the program will be outlined so that it could potentially be assessed in future studies.

There are three key research questions:

1. **RQ 1** Is the accuracy of classifications greater than just a human coder
2. **RQ 2** Is there a reduction in time taken for classifications.

3. **RQ 3** Are the effects (if any) on the accuracy and speed greater or lesser with inexperienced coders compared to experienced coders.

To ensure that these research questions are answered reliable we must design an experiment which maximises external validity. This is so that the experiments can be repeated and the findings validated. With this in mind the experiment will have three main design principles.

- The testing environment needs to as close to the real coding environment as possible. This is so that all results found in experimentation are representative of the applications of the program.
- The test material must be representative of the reports intended to be classified in it's real application. This means that they must vary in content and length and be a representative sample of the entire collection of reports.
- The experiment must allow for feedback from the coders so that we can track and evaluate how the coders interact with the program. This would allow us to make any improvements to the workflow and efficiency of the program.

First the participants must be selected. As the program is specific to LeDeR only previously approved coders may be used. The approval to access the data is limited and as such cannot be widened for experimental purposes. This limits the number of potential participants. We must also aim to satisfy RQ3, the comparison between experienced coders and inexperienced, to be determined by a LeDeR supervisor. The experiment will therefore require a minimum of two "experienced" coders and two "inexperienced" coders. This would help negate the individuals ability to use the program and focus on the difference caused by experience not individual preference. The more participants the more valid the results, however with the limit on personnel for the LeDeR program four participants would be an acceptable minimum.

The experiment will be conducted one participant at a time. The participant will work at a dual monitor set up as close as possible to the ones used in LeDeR. The participant will be accompanied by an observer. The observer's role is to note down and encourage any verbal feedback from the participant and also to answer any questions on the usage of the program. Data will be collected by recording the time taken to complete the report. The observer will also record all verbal feedback to see how the coder interacted with the program so that any improvements could be noted. Finally the screens of the coder will be recorded to see when the program was used and when it was not, and also how it was used. This is so that we can see which classifications it would be most useful for as well as if the inexperienced coder relied on the predictions more than the experienced. We would also be able to tell if the coders did not use it all, some may even find it an inconvenience.

Each participant will code the same five reports as all other participants. This will consist of three separate sets, firstly the training set. The set consists of one shorter introductory pen portrait, this is so that the participants can become used to the program before the test. Data for the training set will also be recorded so that the speed at which participants learn how to use the program can be compared. The second set is the null set, a set of two pen portraits. The participants will code the report in the normal way without the additional classification report, this is so the effect of the program on the coders speed can be quantified. The final set consists of two pen portraits for which the participants code using the program. For all pen portraits they will have been coded before not by the participants, this allows us to test the accuracy of the coding. It should also show if the breadth of the codes has increased. Both the null and

test sets will consist of two reports, a shorter report with fewer labels and a longer report with a greater number of labels. It is vital that shorter reports and longer reports in both the null and test set have similar structure, meaning the label density is similar. Comparisons made between the two sets must be made so the sets must be as similar as possible. This can be measured by label frequency, distribution and token length. All factors that can effect the speed of labelling a document. A further action to increase validity the null and test sets will be randomly selected for different participants so that any differences in reports will have minimal impact on findings.

9.3 Sample Report

Learning Disabilities Mortality Review (LeDeR) Programme

Initial Review

Template Version: IR08

Person ID: 25131109

Region of England: Midlands and East - West Midlands

Date of notification: 18/04/2018

Reviewer name: trainer three

File upload link: https://leder-upload-test.rit.bristol.ac.uk/leder-upload/casefiles?caseId=25131109

How to carry out an Initial Review

Questions 1 – 36 below take information from the death notification.

Please review these and then answer the remaining questions.

Thank you.

Death notification information

1. Name of the person notifying the death

Name: Martine Colshaw

2. Job and employing organisation of person notifying the death (if relevant)

Details: Liason nurse, West Mids Royal Hospital

3. How the reporter knew the person who has died

Relationship: She died in the hospital where I am employed

Reporter's contact details (if they are happy to be contacted)

4. Telephone number: 01902 887556
5. Email address: martinecolshaw3@nhs.net
6. Postal address and postcode: West Mids Royal Hospital
   Great Simmons Street
   Wolverhampton
   WV1 3RD
7. Reporter's preferred method for contact: Email

Please note that from this point forwards, answers shown in red indicate that a full multi-agency review might be required.

8. Who else has been notified about the death? (Tick all that apply)

☒ To the reporter's knowledge, no one else has been notified

☐Coroner ☐Safeguarding Team

☐Child Death Review ☐Police

☐Care Quality Commission ☐Someone else

☐I don't know

If someone else has been notified about the death, please provide their contact details if you have them.

Contact details:

Details about the person who died

9. FIRST (GIVEN) NAME of the person who died

Name: Carrie

10. LAST NAME (i.e. family name or surname) of the person who died

Name: Walker

11. Was the person known by any other name? If so, what was it?

Name:

12. Date of BIRTH (This should be in the format dd/mm/yyyy)

Date: 03/05/1978

13. Date of DEATH (This should be in the format dd/mm/yyyy)

Date: 18/02/2018

14. Age at death

Age: 39

15. Gender

☐Male ☒Female ☐Other

If 'Other', please describe:

16. Deceased person's ethnic group

White

☒British

☐Irish

☐Gypsy or Irish Traveller

☐Any other White background (please give details in box below)

Mixed / multiple ethnic groups

☐White and Black Caribbean

☐White and Black African

☐White and Asian

☐Any other Mixed / multiple ethnic background (please give details in box below)

Asian / Asian British

☐Indian

☐Pakistani

Black / African / Caribbean / Black British

☐African

☐Caribbean

☐Bangladeshi

☐Any other Black / African / Caribbean background (please give details in box below)

☐Chinese

☐Any other Asian background (please give details in box below)

Other ethnic group

☐Arab

☐Any other ethnic group (please give details in box below)

Details of person's ethnic group:

17. Marital Status of the person who died

☒Single (never married)

☐Married / civil partnership

☐Separated (but still legally married / in a civil partnership)

☐Divorced

☐Widowed

☐I don't know

18. In which area of England was the person registered with a GP?

☐ North: Yorkshire & the Humber

☐North: Lancashire

☐North: Greater Manchester

☐North: Cumbria & the North East

☐North: Cheshire & Merseyside

☐Midlands & East: North Midlands

☐Midlands & East: Central Midlands

☒Midlands & East: West Midlands

☐Midlands & East: East Midlands

☐South: South West

☐South: South East

☐South: Wessex

☐South: South Central

☐London Region

☐I don't know

☐Not registered with a GP

19. NHS Number (This should be in the following standard format: 000 000 0000)

NHS number: 767 554 8941

20. Did the person who died have any known medical conditions or health problems?

Details: Down's syndrome, obesity, recurrent cellulitis, eczema, HO DVT

21. What level of learning disability did the person who died have?

☐Mild ☒Moderate

☐Severe ☐Profound / multiple

☐I don't know

22. Usual address and postcode of the person who died

Address: 4 Barley Close, Wolverhampton,

Postcode: WV5 6TY

23. Did the person who died usually live alone?

☐Yes ☐No ☒I don't know

24. Was the person who died placed out-of-area, either in a residential / nursing placement or in a supported living tenancy?

☐Yes ☒No ☐I don't know

If yes, please state which area was their original 'home':

25. Was the person subject to any restrictive legislation?

☐None

☐Deprivation of Liberty Safeguards (DOLS) - approved

☐Deprivation of Liberty Safeguards (DOLS) – applied for

☐Section of the Mental Health Act

☐Detention in police custody/imprisonment

☐Other:

☒ don't know

If the person was subject to any restrictive legislation, please describe more fully (e.g. dates, reason for restriction)

26. Someone who knew the person who died

Name: Sandra Denmark

Telephone number: 01902 347673

Email address: sandydennymaybe@googlemail.co.uk

Address and postcode: 142 Turner Drive, Wolverhampton, WV3 7CD

27. How did they know the person who died?

Sister

28. Name of and contact details of the person's GP surgery

GP name: Dr DC Brockley

Surgery contact details: Kathryn Buildings Surgery, 455 Mayflower Road, Wolverhampton, WV5 3WA

Details of the Death

29. What was the place of death?

☒Hospital

☐Usual place of residence

☐Hospice / palliative care unit

☐Home of relative or friend

☐Residential / nursing home that was not usual address

❑ don't know

☐Other:

If the person died in hospital, please state the name of the hospital, NHS Trust or Foundation Trust (if known):

Royal Mids Hospital

30. What was the cause of death as described on the Cause of Death Certificate? (If you do not know, please leave blank)

I	(a) Disease or condition leading directly to death	Pulmonary Embolus
I	(b) Other disease or condition, if any, leading to I(a)	Deep Vein Thrombosis
I	(c) Other disease or condition, if any, leading to I(b)	
II	Other significant conditions contributing to death but not related to the disease or condition causing it	Cellulitis Obesity

31. What did reporter think the cause of death was?

Perceived cause: As above

32. Will there be a post mortem?

☐Yes ☒ No ❑ don't know

33. Will there be a Coroner's inquest?

☐Yes ☒No ❑ don't know

34. Will there be any other investigation or review of the death?

☒Yes ☐No ❑ don't know

If YES please describe: SJR

35. Reporter's comments about the death:

Patient was not on an acute ward at time of death, she was having a period of rehab before discharge home to ensure she could look after herself

END OF DEATH NOTIFICATION

Initial Review Of Death – additional questions

In preparation for the initial review of the person's death, please:

- Check and complete the information received at notification.
- Contact the notifier/s to ensure their views are included in this review.
- Identify someone who knew the person well (e.g. close family member) and speak to them about the person themselves and the circumstances leading to their death. Ask them to help you complete a pen portrait of the person who has died, and a timeline of the circumstances leading to their death.
- Review at least one set of relevant case notes (e.g. hospital record, electronic GP summary record, social care record).

In order to upload case review notes from agencies, please contact the individuals involved and ask them to use the following link. When they click on this link they will be able to upload files which you can access from the LeDeR dashboard.

File upload link: https://leder-upload-test.rit.bristol.ac.uk/leder-upload/casefiles?caseId=25131109

36. Please use this space to write your own notes, comment or thoughts about this review. (It is useful for us to be able to see your train of thought. However, you are welcome to delete these prior to submitting your completed review if you wish to do so.)

37. Please check the answers to notification questions and complete/amend where necessary (there should be no 'I don't know' answers at point of submission).

☒Please tick to confirm that all notification questions (except Q31) are complete and correct

38. It is an integral part of the LeDeR methodology that family are given the opportunity to share their experiences and any learning they would like to pass on. Have involved family member/s been given the opportunity to contribute to this review?

☒Yes ☐No ☐N/A (There are no involved family members)

If not, please explain why:

39. Please list who has provided information (name and role) about the person and the circumstances leading to their death.

Sister and brother-in-law

40. Please describe what case notes you have reviewed:

MYHT secondary case notes last 3 attendances 12 months OP Christchurch Care support notes

41. How was the person's care funded? (Please tick all that apply)

☒Local Authority (directly commissioned)

☐Local Authority via direct payment / personal budget

☐NHS (directly commissioned)

☐NHS via personal health budget

☐CHC (Continuing healthcare) funding

☐Joint-funding (NHS and local authority)

☐Section 117 aftercare arrangements

☐The person, or their family, themselves

☐Other:

Please add any comments about this here:

42. Was the person who died in regular contact with any of the following?

☒Family member(s)

☐An attorney under a Lasting Power of Attorney direction

☐A Deputy agreed / appointed by the Court of Protection

☐An advocate

☐Other:

Please add any further details:

43. Did the person who died usually receive support from a paid support worker from outside their family (whether from the private, statutory or voluntary sector)?

☒Yes ☐No

If YES, did they receive support:

☒Daytime only

☐Day and night (waking night)

☐Day and night (sleeping night)

Please describe any services and support that the person received:

Carrie recieved 30 mins a day morning support and 30 mins evening for medication, admin and general wellbeing check. Support to shop for one hour twice a week

44. Did the person who died experience any of the following changes in service provision in the past year?

☐Yes, change in service PROVISION (e.g. hours of support)

☐Yes, change in service PROVIDER

☐Yes, change in PLACE of provision

☐Yes, leaving the care of the local authority

☒ No

☐Not in receipt of services

If YES can you provide details (e.g. why provision changed, number of changes, what changes were made, impact of changes):

45. Did the person have a Learning Disabilities Annual Health Check in the last 12 months?

☒Yes ☐No ☐Not known

If no, please explain why not:

46. Was the person on an end of life pathway?

☐Yes ☒No

If yes, please add any comments about this here:

47. **Pen portrait of the individual**

Please include the following:

- Information about the person and their health
- Information about the environment in which the person was living
- Description of support arrangements (if any)
- Contact with services (primary care, secondary care, specialist learning disability, social care, other)
- Coordination of care

You can find guidance about writing a pen portrait in the 'Help' section on your LeDeR dashboard.

Carrie had Downs syndrome and a moderate learning disability. She was born and lived in Wolverhampton all her life. Carrie had an older sister, Sandra (Sandy) and throughout her life she and Sandy were very close, perhaps because their mother died while they were very young (Carrie was three).

Following Carrie's father having a stroke at the age of 60, Carrie also became a carer. Carrie was able to run errands; supporting her dad to maintain the house and go to the local shops with a list. Sandy visited daily to cook and complete other tasks. Six months before their dad died he needed to go to nursing care and Carrie, then aged 25, went to live with Sandy and her family where she remained for about

10 years, until she moved to independent living in 2012. In her mid 30s it was noticed that Carrie had some difficulty seeing smaller items, and visited an optician in who had a daughter with learning difficulties. Tests showed that Carrie had always had defective vision in one eye, and she was delighted when she received glasses to be able to read bus numbers at a distance. Carrie would also have benefited from wearing a hearing aid however she didn't want to wear one as she associated this with being old and she always said she never wanted to get old.

Carrie was described as a resourceful person and travelled independently by bus and train; if all else failed and Carrie missed her bus she would catch a taxi or even charm her way to getting a lift from someone she knew.

For most of the 10 years with Sandy and her family, Carrie always expressed a wish to stay with them. However, when the older children became teenagers, Carrie struggled with sharing with them and started to think about moving out and getting her own place; Sandy and her family supported Carrie to prepare for this next step by allowing her to stay at their home by herself when they went away for a few days, making sure she had food she could prepare. Carrie also attended cooking sessions and was supported by the Occupational Therapist of the local CLDT to develop her independence skills.

Carrie met her goal to live independently and moved to Barley Close in 2012. Barley Close is a purpose built supported living service made up of individual flats where people receive the assessed support they required from Christchurch Care. Sandy said she was very relieved that they were able to get a place for Carrie at Barley Close as she knew that Carrie would have difficulties living in a shared house due to her strong character but benefited from staff and company on site. Sandy said that Barley Avenue was just the right set-up for Carrie and Christchurch Care staff were 'fabulous people'.

Carrie struggled with following a healthy lifestyle and making good food choices and did not like to exercise. She benefited from taking part in healthy eating sessions run by the learning disability community nursing service in 2011 and receiving support from Christchurch Care staff to complete her shopping twice a week. Sandy would have liked to see Carrie receive more support in this area as she said Carrie was receptive to support to prepare healthier food but without that would eat mostly fast food. Carrie struggled to keep her weight under control throughout her life and around the time of her death she weighed around 19 stones.

During her life Carrie had an eventful love life with many boyfriends and she also had lots of friends. She was a very easy person to be around and her friends enjoyed her company. Carrie enjoyed listening to pop music and she was a great dancer; she loved to be helpful and to know other people's business and referred to herself as "the top lady at Barley Avenue". She is very sadly missed by her family and friends

especially as her death came as a great shock.

Carrie had a history of a previous DVT in 2008 and was prescribed Warfarin as a result. However she came off Warfarin in 2014. Her family described Carrie as having a very high tolerance to pain. Carrie's family shared that in January and February 2018 she had 3 admissions to ED in a short time and was identified as having a learning disability on the 3rd presentation which resulted in an admission. Carrie was admitted to the Acute Assessment Unit in Royal Mids hospital and then to Royal Mids Intermediate Care Unit for rehab where she died. Carrie's family feel the recent DVT may have been missed and diagnosed as an infection in her leg; they were also unsure if a DNACPR notification was in place and if CPR took place during the last episode of care in hospital? After reviewing the hospital notes I was able to confirm with Sandy that a DNACPR notification was not in place for Carrie and CPR took place but was unsuccessful.

Carrie had 3 attendances at ED in and 1 admission at which she was treated for infection in her lower right leg (Cellulitis). Despite previous history of DVT a Wells score was not completed in ED on the 13th of February however from admission she was prescribed a prophylactic dose of Deltaparin and assessed as requiring a period of Rehab at PICU due to immobility and living alone; only receiving support at key times during the day. Carrie wanted to return home to her flat and her job at the market. She did not want to use any walking aids and stated she didnt want to get old. She was seen by the ALN for LD in the acute trust who recommended rehab and that she could not go home if unable to mobilise.

Carrie had an extended period 4 years of Warfarin therapy post DVT in 2008 - unsure why in 2014 this was discontinued.

48. What prescribed medications did the person usually take?

To add more rows, click into the last row of the table, right click, select Insert – Insert Rows Below.

Usual medications			
Medication name	Dose	Frequency	Purpose

What prescribed medications were they taking at the time of their death?

Medications at the time of death

Medication name	Dose	Frequency	Purpose

What was the date of the last medication review? (This should be in the format dd/mm/yyyy).

Date: ☒Not known

Timeline for circumstances leading to death

To add more rows, click into the last row of the table, right click, select Insert – Insert Rows Below.

N.B. The person's death should be the last line in the timeline.

Date **(from earliest to latest)**	**Reported by / where evidence obtained from**	**Circumstances**
08/10/2012	MYHT secondary care notes	DVT confirmed by ultra sound scan DVT located in Long Saphenous vein. Treated with anticoagulent therapy.
04/7/2017	MYHT secondary care notes	Attended ED at PGH ? DVT D-Dimer blood test returned positive. Deltaparin anticoag injection administered. Appointment made to attend Ambulatory Care next day (05/05/2016) ultra sound scan which returned negative. ??? Need to repeat scan x 1 week.
16/10/2017	MYHT secondary care notes	Attended ED DVT D-Dimmer blood test and ultra sound scan completed both returned negative. Diagnosed skin infection.
05/11/2017	Christchurch Care support records	GP appointment to discuss skin condition (both legs, arms, back) awaiting OP appointment with Dermatology.
13/12/2017	Christchurch Care support records	Annual Health Check at GP BP and Pulse checked – slightly raised (140/90mmHg). Flu and Pneumonia Jab administered. Routine blood sample taken. Discussed skin condition cream prescribed (Fucidin) Weight management plan in place with monitoring to take place in primary care.
21/12/2017	MYHT secondary care notes	Dermatology appointment at PGI pressure stockings prescribed also prescribed antibiotics.
29/01/2018	MYHT secondary care notes	Collapsed whilst working at the market ambulance crew contacted carers for advice?

		Faint? Seizure? Ongoing cellulitis. Shortness of breath? Wheeze. History taken re DVT
01/02/2018	Christchurch Care Medication charts	zeroderm ointment, zerobase cream, Donepezil 10mg Tab, Clobestasone Ointment, Desunin Tab, Atorvastatin
10/02/2018	MYHT secondary care notes	Attended ED at PGH supported by a member of her support team from Christchurch Care. Diagnosis ongoing Cellulitis, history of frequency of Micturation. History of DVT identified. No Wells assessment undertaken
12/02/2018	MYHT secondary care notes	Carrie attended ED at PGH (16.19) with no staff support seen by FY2 (20.40) Admitted under medicine as a failed discharge. Ambulance impression infection
12/02/2018	MYHT regular Prescriptions	Dalteparin 5000unit, Donepezil 10mg, Atorvastatin 20mg, Colecalciferol 4000 unit, Paracetamol 1 gram, Flucloxacillin 500mg
13/02/2018	MYHT secondary care notes	Patient was prescribed and administered Deltaparin throughout hospital admission (14,15th, 16,17th Nov 2016).
13/02/2018	MYHT secondary care notes	Physiological observations completed as required and stable throughout the day prior to PICU transfer- no documented evidence of clinical deterioration of the patient's condition prior to PICU transfer
17/02/2018	MYHT secondary care notes	Full medical consultant and physiotherapy review on the day of patient transfer and clear documentation deeming the patient stable for transfer by both specialists.
17/02/2018	MYHT secondary care notes	On transfer to PICU patients NEWS 0 and documentation states the patient is comfortable and stable on arrival.
18/02/2018	MYHT secondary care notes	Moved to PICU (Intermediate Care Unit) Rehab plan in place. Unable to weight bear hoist x2 goal to mobilise with Zimmer frame with the support of 1
18/02/2018	MYHT secondary care notes	Patient cardiac arrested at 21:30 after becoming SOB on mobilising to the bathroom- CPR commenced. CPR stopped at 22:30. Coroner report state cause of death is: PE, DVT and cellulitis

49. Has anyone expressed any concern about this death?

☒Yes ☐Not to my knowledge

If yes, please add any comments about this here: Family members whilst not wanting to make a formal complaint about Carrie's care and treatment in Royal Mids

feel there may have been a missed DVT which could have been identified and treated

50. Did the person have a Do Not Attempt Cardiopulmonary Resuscitation (DNACPR) order in place at the time of their death?

☐Yes ☒No ☐ don't know

If YES, was the documentation correctly completed and followed? (Please tick only one option)

☐The DNACPR documentation was correctly completed and followed

☐The DNACPR documentation was correctly completed but was not followed

☐The DNACPR documentation was neither completed nor followed correctly

☐ don't know

Please add any comments about this here:

51. Please describe any decisions where there is evidence that a mental capacity assessment took place and, if indicated, a best interests decision-making process was followed:

(Please use a separate line for each decision)

To add more rows, click into the last row of the table, right click, select Insert – Insert Rows Below.

Decisions
Last 3 attendances at hospital were at ED and diagnostic in nature. Carrie was able to give consent to rehab care plan

Please describe any decisions around which you think a mental capacity assessment and best interests decision-making process should have taken place but did not.

Decisions

Please add any comments about the use of the Mental Capacity Act here:

52. Please describe any reasonable adjustments that were provided for the person:

(Please use a separate line for each one)

To add more rows, click into the last row of the table, right click, select Insert – Insert Rows Below.

Adjustments

Please describe what reasonable adjustments should have been provided but were not.

Adjustments

Please add any comments about the provision of reasonable adjustments here:

53. From the evidence you have, do you think that there were delays in the person's care or treatment that adversely affected their health?

☐Yes ☒No

Please add any comments about this here: The care received for Carrie's last admission, particularly the holistic care, physio referral and transfer for rehab was good practice.

54. From the evidence you have, do you think that this death might be attributable to abuse or neglect in any setting?

☐Yes ☒No

Please add any comments about this here:

55. From the evidence you have, do you think that problems with organisational systems and processes (including the coordination of care) led to a poor standard of care?

☐Yes ☒No

Please add any comments about this here:

56. From the evidence you have, do there appear to be any gaps in service provision that might have contributed to the person's death?

☐Yes ☒No

Please add any comments about this here:

57. To your knowledge, is there evidence that the person, in the 12 months prior to death, was subject to:

Significant and / or continuing safeguarding concerns:

☐Yes ☒No

A safeguarding plan:

☐Yes ☒No

Please add any comments about this here:

58. After reviewing this death, have you identified any best practice?
N.B. 'Best' practice refers to that which is over and above the standard of care that should be usually be expected.

☐Yes ☒No

Please add any comments about this here:

59. After reviewing this death, do you think that any further learning could be gained from a multi-agency review of the death that would contribute to improving practice?

☐Yes ☒No

Please add any comments about this here:

60. From the information that you have, and in discussion with your local area contact, please grade your overall assessment of the care received by the person on a scale of 1 (best) to 6 (worst):

☐1. This was excellent care (it exceeded current good practice).

☒2. This was good care (it met current good practice in all areas).

☐3. This was satisfactory care (it fell short of current good practice in minor areas, and no significant learning would result from a fuller review of the death).

☐4. Care fell short of current best practice in one or more significant areas, but this is not considered to have had the potential for adverse impact on the person and no significant learning would result from a fuller review of the death.

☐5. Care fell short of current best practice in one or more significant areas, although this is not considered to have had the potential for adverse impact on the person, some learning could result from a fuller review of the death.

☐6. Care fell short of current best practice in one or more significant areas resulting in the potential for, or actual, adverse impact on the person.

Please explain your reasons for giving this grade: Annual Health Check with outcomes and follow up completed in primary care.

Despite no Wells assessment completed at last attendance at ED 13th of November and treated for infection Carrie received Deltaparin medication 14,15,16,17 of November. The last admission DVT was not suspected, she was treated for Cellulitis.

Note: If you have insufficient information and are unable to grade your overall assessment of the care received, please seek further information until you can do so - for example, review further case notes or speak to those who knew the person well.

61. Please add any additional comments you might have in relation to this review (e.g. any particular difficulties you have had in completing this review).

Additional Comments: Difficulty locating all secondary care notes made the review more difficult to complete and less robust only ED and Rehab care plan available in secondary care.

62. Please add any comments that you might have about your experience of the LeDeR Review process or IT System.

Comments:

Next Action

63. Please review the options below and select one to decide your next action.

1. **If you have answered any questions with an answer that is coloured red**, a multi-agency review of this death should be considered.

☐ Please tick box if this applies

2. **If this person meets the criteria for the current priority themed review** deaths (the person was aged 18-24 (inclusive) when they died, or they came from a black or minority ethnic background), a multi-agency review of the death is required.

☐ Please tick box if this applies

3. If your initial assessment of this death suggests that NO multi-agency review is required, but you think that such a review might be appropriate, (i.e. further learning could be gained from a multi-agency review of the death that would contribute to improving practice), please do conduct a multi-agency review.

☐ Please tick box if this applies

4. If your initial assessment of this death suggests that NO multi-agency review is required, and you consider that no further learning could be gained from a multi-agency review of the death that would contribute to improving practice, please complete the Action Plan below and submit the Initial Review and Action Plan to your local area contact.

☒ Please tick box if this applies

Thank you.

YOUR NEXT ACTION:

Please now either START A MULTI-AGENCY REVIEW

Or

Note any learnings and recommendations that have come from this review in the tables below then submit this document to your local area contact.

Identified Issue	**Learning**	**Recommendation to address issue**
e.g. Zack was discharged from hospital without the care home staff being trained in catheter care which led to him having a UTI.	*e.g. Nursing staff do not routinely assess specific skills of care home staff before discharge.*	*e.g. Skills/capabilities of care home staff must be assessed prior to discharge and necessary training given before patient is discharged.*

Any best practice identified:

Best practice identified	**Any recommendations or learning from this**

FINALLY:

Has the family given consent for information about the review to be shared in order to improve service provision if appropriate?

☐Yes ☐No

If yes, please upload the consent form to the LeDeR web-based platform, or post to the LeDeR team, before submitting this review.