THE UNIVERSITY OF QUEENSLAND

AUSTRALIA

Text Analytics for the Queensland Ambulance Service

by

Jiayong Kuang

School of Information Technology and Electrical Engineering,

University of Queensland.

Submitted for the degree of Master of Data Science

11th, November, 2019

Jiayong Kuang

jiayong.kuang@uq.net.au

11th, November, 2019

Prof Amin Abbosh
Acting Head of School
School of Information Technology and Electrical Engineering
The University of Queensland
St Lucia QLD 4072

Dear Professor Abbosh,

In accordance with the requirements of the Degree of Master of Data Science (Coursework) in the School of Information Technology and Electrical Engineering, I submit the following thesis entitled

"Text Analytics for the Queensland Ambulance Service"

The thesis was performed under the supervision of Mark Griffin. I declare that the work submitted in the thesis is my own, except as acknowledged in the text and footnotes, and that it has not previously been submitted for a degree at the University of Queensland or any other institution.

Yours sincerely

Jiayong Kuang

Abstract

Car accidents are prevalent at present and cause severe losses every year (In 2016, the total costs came in over $6.2 billion in Queensland). To explore the causes of car accidents, the patients' condition and the treatments used in each car accident case in Queensland, text analysis was used to analyze the car collision data of the Queensland Ambulance Service. This resulted in a geographic map of car accident frequency and density. Moreover, the cause of road accidents can be extracted from the reports using a developed machine learning workflow automatically. This project will help us to understand the circumstances of the accidents (what happened and why).

Content

1.Introduction

1.1 Background

The number of hospitalizations and deaths resulting from road accident injuries has remained stable at a high level (more than 6000 hospital casualties each year) in Queensland from 2012 to 2017 (Queensland Road Crash Report, 2019). Moreover, the Queensland Ambulance Service (QAS) (which provides patient transfer and treatment services) visits approximately 700,000 incidents per year. This large number may be because people have not pursued the causes of accidents or taken adequate measures to reduce its incidence for many years. In addition, with an insight into the Queensland car accident map, QAS can allocate ambulances and medical resources more appropriately to improve response efficiency and service quality. Therefore, I propose to use data science methods to analyze Queensland Ambulance Services data (1 January 2015 - 30 November 2018) to find out the reason why accidents keep frequently happening and to suggest strategies on how to prevent them or reduce the damage as much as possible.

1.2 Queensland Ambulance Service

The QAS provides medical care and patient transfer services for an estimated five million Queensland inhabitants that are geographically scattered over 1.7 million square kilometers (Queensland Ambulance Service, n.d.). Different from other Australian states, Queensland has a high proportion of its population living in regional or remote areas (37%, compared with 21-26% in other states). In the QAS official website, they also indicate that QAS operates as a statewide service within Queensland Health and is accountable for the delivery of pre-hospital ambulance response services, emergency, and non-emergency pre-hospital patient care and transport services, inter-facility ambulance transport, casualty room services, and

planning and coordination of multi-casualty incidents and disasters(Queensland Ambulance Service Local Ambulance Networks, Queensland Government). The QAS delivers ambulance services from 296 response locations through 15 Local Ambulance Service Networks (LASNs), that are geographically aligned with Queensland Health's Hospital and Health Services' boundaries. The QAS has a 16th statewide LASN which comprises eight operations centers distributed throughout Queensland that manage emergency call taking, operational deployment, dispatch, and coordination of non-urgent patient transport services. The QAS is organized into fifteen geographical Local Ambulance Service Networks (LASNs): Torres and Cape, Cairns and Hinterland, Townsville, Mackay, North West, Central West, Central Queensland, Wide Bay, South West, Darling Downs, Sunshine Coast, Metro-North (Brisbane), Metro-South (Brisbane), West Moreton, Gold Coast.

1.3 Data Description

The data record all the car accidents the Queensland Ambulance Service responded from 1 January 2015 to 30 November 2018. The data contain about 9,1384 records. In these records, 6,1620 of them are cause-unclassified car collision reports, and the remaining 2,9764 are cause-classified truck collision reports. The project aims to use all the records to identify the accident distribution and use the cause-classified records to develop a machine learning procedure to classify the causes of accidents.

There are a few variables in the dataset, including report date, case nature, accident location, final assessment, patient age, ambulance response date and time, and patient gender. In this case, the case nature variable indicates what the paramedic believes is the cause of the presenting problem. Moreover, the final assessment describes what the paramedic believes is the patient's primary problem at the time the patient is discharged from his/her care (Queensland Ambulance Service Application for Data form, n.d.). Also, the accident location includes the suburb, postcode, and street name.

Therefore, an in-depth analysis of the data to discover the frequency of traffic accidents in various regions, the relationship between accidents and geographical distribution has been conducted in the last semester. And this semester, the project focused on the classification of the reports.

1.4 Challenges

The Queensland Ambulance Service data is hard to classify, there are three main reasons for this. Firstly, all the car accident reports are all organized in a similar way and describe a homogeneous topic related to the car accident and patient treatment. A considerable number of specific words are used frequently in all the reports, for example, pt (patient), car, pain, and nil. Therefore, normal classification methods (including the LDA topic model) do not satisfactorily classify the reports. Secondly, the QAS reports are neither standardized in writing nor sometimes organized in natural English due to a type of shorthand. In some cases, the reports used word combinations to represent specific meaning instead of using sentences. Thirdly, the QAS data used several different abbreviations to represent the same term. For example, years old might be written as “yo”,”y.o”,”yrs”,and “yro”. Thus, some classification methods might not be able to identify the homogeneity of these abbreviations.

2. Theory

In the previous step (data cleaning and general analysis), I applied text analysis methods using R to analyze the reports. First, all the words in the report were retrieved and the meaningless words were eliminated. Second, the word frequency and word correlation were calculated to identify the accident-prone suburbs and streets. The accident distribution was visualized on the Queensland map. Third, LDA language models using different parameters were used to classify the reports. As a

result, the data has been cleaned comparatively. Furthermore, most of the information in the reports can be converted into structured data and analyzed using machine learning methods.

In the second step (data modeling and text classification), to classify the reports efficiently, some machine learning methods have been used according to the peculiarity of the QAS data. First, based on the cleaned data, a term-frequency inverse document-frequency matrix was used to represent the frequency of each unique word in each document and a classifier was built based on it. Second, the part of speech of each word was tagged and the keywords in each sentence were identified. Using the result, another matrix representing the frequency of each unique keyword in each document was generated and used to build a classifier to classify the reports. Third, a vector matrix was developed to represent all the unique words in the reports using deep learning algorithms. A classification model was built using this word vector matrix to represent reports. As a result, the most suitable method for QAS data for the next step in the analysis can be determined by comparing the performances of these three classifiers.

2.1 Text Data Cleaning and Text Mining Technique

Text mining (also called text analytics) was used to analyze the QAS data. It is the basic process of extracting high-quality information from unstructured text (Hearst, 2012). Text analytics usually involves the operation of structuring text, analyzing text patterns, evaluating and converting text. Generally, people would use four analytics methods for text analytics. The first step is tokenization. This is the process of dividing and possibly classifying parts of input strings. The second step is the term frequency analysis. This is a method intended to reflect how important a word is to a document in a collection. In this process, stop words, which are also regarded as common but useless words (such as his, is, the), would be removed from the tokens. The third step is the word correlation analysis. In this step, the correlation coefficient

among the tokens will be computed as a matrix. Therefore, the correlation coefficient will be a measurement to demonstrate how often the words appear together in the same record. The fourth step is topic modeling. Topic modeling is statistical modeling used to discover the abstract “topics” that appear in a collection of documents. There are a few topic models based on different algorithms that can be used to classify individual text into a specific topic. Finally, all records will be given one or multiple labels based on their topics (Zhang & Wang, 2010).

2.1.1 Tokenization and Stemming

Tokenization is the process of segmenting string series into chunks like letters, terms, sentences, and symbols. The generated results of this process are called tokens. Tokens can be letters, phrases or even sentences in their entirety based on user definitions. Many characters such as punctuations are discarded in the tokenization process. Tokens then become the input of stemming. Throughout computer science, tokenization is used where it plays a large part in the human-machine language interpretation process (Method for language-independent text tokenization using a character categorization, 1991).

Stemming the context retrieved in the tokenization step can avoid the words with the same root and similar meaning being recognized as different words. In text analytics, stemming is the process of turning inflected (or sometimes derived) words into their word stem, base, root, or the most commonly used form (Lovins, 1968). The stem need not be indistinguishable to the morphological foundation of the word; it is typically adequate that related words guide to a similar stem, regardless of whether this stem is not in itself a substantial root. Calculations for stemming have been contemplated in software engineering since the 1960s. Many web search tools treat words with a similar stem as equivalent words in a sort of inquiry extension, a procedure called conflation. Stemming can significantly reduce the workload of text analytics and reduce computation time.

2.1.2 Removing the Stop Words

Stop words are words commonly used (such as “the,” “a,” ”an,” “his”) where there is no significant meaning that can be used to analyze their frequency or distribution in a text (Rajaraman & Ullman 2011). Thus, search engines have been programmed to ignore all the stop words for both searching queries and results. These words would take up space, valuable processing time, and not be relevant to the modeling topic. For this, the stop words need to be removed from the tokens.

2.1.3 N-Grams

In the fields of natural language processing, an n-gram is a contiguous sequence of n items from a given sample of text. The items can be words, letters, numbers, syllables or base pairs according to the application. The N-Grams are collected from a text corpus. When the items are words, N-Grams may also use Latin numerical prefixes in their representation (Schonlau and Guenther, 2016). For example, an n-gram of size one is referred to as a "unigram"; size 2 is a "bigram"; size 3 is a "trigram."

N-Grams models are widely utilized in statistical natural language processing. In intelligence assistance and searching engine queries processing, words and sequences of words are displayed utilizing an n-gram distribution. For constituency parsing, words are modeled such that each n-gram is composed of n words. The sequences of words are considered as an n-gram in this project. For sequences of words, the trigrams can be generated from "the girl opened the box" is "the girl opened," "girl opened the," and "opened the box."

2.1.4 TF-IIDF

In information retrieval, TF-IDF or term frequency-inverse document frequency, is a numerical index that is intended to reflect how important a word or a term is to a document in a collection or corpus (Frequency analysis, 2019). It is widely used as a weighting factor in topic modeling. The TF-IDF index is relative to the number of times a word shows up in the document and is balanced by the number of archives in the corpus that contain the word. This measurement accounts for the fact that some words appear more frequently in general. TF-IDF is the most widely used method for word frequency analysis and term-weighting schemes today.

2.1.5 Terms Correlation

Term correlation refers to the co-occurrence of terms. There are two different methods to determine whether two terms are correlated. The first one is to calculate the conditional probability based on the Bayesian formula. For example, if a term appears (or appears at a high frequency), and another word has a high probability of appearing (or appearing at a high frequency) in the same corpus, then there is a strong correlation between them. The second method is to compute the phi coefficient, a common measure for binary correlation. This project will use this method. It focuses on how much more likely it is that either both two terms appear or neither, rather than that only one term appears without the other.

	Has word Y	No word Y	Total
Has word X	n_{11}	n_{10}	$n_{1\cdot}$
No word X	n_{01}	n_{00}	$n_{0\cdot}$
Total	$n_{\cdot 1}$	$n_{\cdot 0}$	n

Table 1. Illustration of Word Pairwise Correlation

For this case, consider n11*n00 as the case terms appear together or neither. n10*n01 is the case that only one term appears without the other (Julia & David, 2019). Hence:

$$\phi = \frac{n_{11}n_{00} - n_{10}n_{01}}{\sqrt{n_{1\cdot}n_{0\cdot}n_{\cdot 0}n_{\cdot 1}}}$$

The correlation probabilities among the tokens will be computed as a matrix for each report. Therefore, the correlation will be a measurement to demonstrate how likely the words appear together in the same record.

2.1.6 Topic Modeling

In natural language processing, a topic model is a type of statistical model for discovering the dynamic "topics" that occur in a collection of reports (Julia & David, 2019). Topic modeling is a frequently used text analytics tool for the discovery of hidden semantic meanings and indications in a text body. Instinctively, given that an archive is about a specific topic, a specific term would be expected to show up in the report more or less frequently: "crash" and "bleed" will appear more often in documents about accidents, "heart attack" and "treatment" will appear in documents about disease, and "patient" and "ambulance" will appear equally in both. Of course, in most of the cases, there would be some overlap between both. Thus, a document can concentrate on multiple topics in different proportions; for example, if there is 10% about patients and 90% about diseases, there would probably be about nine times more disease words than patient words. A critical method for labeling "topics" is clustering related words. A topic model captures this instinct in a mathematical structure based on the statistics of the words in each document. This setting allows the examination of a set of archives and identification of each document's topics and its priority of topics.

In this project, the model latent Dirichlet allocation (LDA) will be used in the topic modeling step. In LDA, each document can be viewed as a mixture of various topics, while each topic is considered as a mixture of words assigned by LDA. LDA is a generative statistical model that enables unobserved groups to explain why some

parts of the data are similar in the observation sets. For example, given word observations collected from archives, it considers that each report is a mixture of a few topics and that each word's essence is attributable to one of the report's topics (Latent Dirichlet allocation, 2019).

Similarly, Probabilistic Latent Semantic Analysis (PLSA) has the same core except that the subject distribution is assumed to have a sparse Dirichlet prior before LDA modeling. The sparse Dirichlet a priori coding intuition, that is, the document covers only a small number of topics, and the topic only uses a small set of words. In practice, this can better disambiguate and more accurately assign documents to topics. LDA is a simplified model of the PLSA, which is equivalent to LDA in the uniform Dirichlet prior distribution.

2.2 Natural Language Processing literature view

2.2.1 Natural Language Processing Introduction

Natural Language Processing (NLP) is a cross-field related to linguistics, computer science, software technology, artificial intelligence, and machine-to-human (natural) language interactions. It is the process of how computers are trained to handle and interpret vast amounts of natural language data. Today, natural Language Processing has overcome many of the obstacles in communication between people and machines by converting natural language to artificial language and providing opportunities for people to perform previously impossible tasks (Pereira and Grosz, 1994). Due to the crucial need to interpret language, NLP has become increasingly important, with its evolving form, implicit meanings, emotions, and purpose. NLP has been widely applied in human lives. The Optical Character Recognition (OCR), Speech Recognition, Machine Translation, and Chatbots are all well-known NLP implementation fields.

In NLP, the learning algorithms accept an input of millions of samples of human-written text (sentences, phrases, and paragraphs). The learning algorithms gain an understanding of the "meaning" of human speech, reading, and other communication methods through observing these examples. Generally, there are five phases of NLP, including lexical (structure) analysis, parsing, semantic analysis, discourse integration, and pragmatic analysis. Most natural language projects use the five-step procedure in a very satisfactory manner. In my project, I did not use each of these five steps though they could have all been useful individually (Arumugam, 2019). To build an appropriate classifier for the Queensland Ambulance Service data, because of the peculiarity of the data, this project mainly aims at lexical analysis and constituency parsing in this step. Additionally, other steps will still be introduced briefly in this section.

2.2.2 Lexical Analysis

The first component in NLP is the lexical analysis (tokenization) (Text Summarization, 2016). This is the method of translating unstructured text documents (such as web pages) into a standardized token set (single words or multiple strings). The Lexical analysis aims to differentiate the theoretically organized phrases into strings. The following tokens will be passed on to the constituency parsing step as an input.

2.2.3 Constituency Parsing

Generally speaking, NLP constituency parsing is a deep parsing method, which is widely used for evaluating a text's syntactic meaning by examining its constituent words based on an underlying (language) grammar. Different to shallow parsing, this not only results in Part of Speech tagged sentences, but also a parse tree showing the syntactic relationship between words, which may also include semantics and other details. For a syntactically ambiguous source, a parse forest or a list of parse trees might be produced based on NLP constituency parsing algorithms (Anderson and Vilares, 2018).

As a general example, two parse tree structures for two sentences from QAS data are listed here.

- The first sentence is “Pt (patient) able to mobilise independently to QAS vehicle.”
- The second sentence is “Pt (patient) has aches to shoulder, flank, lowerback, foot.”

```
[[1]]
(TOP
  (NP
    (NP (NNP Pt))
    (ADJP (JJ able) (S (VP (TO to) (VP (VB mobilise) (ADVP (RB independently)) (PP (TO to) (NP (NNP QAS) (NN vehicle)))))))
    (. .)))

[[2]]
(TOP
  (S
    (NP (NNP Pt))
    (VP
      (VBZ has)
      (NP
        (NP (NP (NNS aches)) (PP (TO to) (NP (NNP R) (NN shoulder))))
        (, ,)
        (NP (NNP R) (NNP Flank))
        (, ,)
        (NP (NN R) (NN lowerback))
        (, ,)
        (NP (NNP R) (NN foot))))
    (. .)))
```

Figure 1. Example of Constituency Parsing from QAS Reports (Generated by R)

As shown in Figure 1, this technique can not only tag the part of speech of each word in the sentence but also build the structure of the sentence to identify nouns described by the same verb. In this example, both shoulder, flank, lower back, and foot have aches. As in the result, they are both tagged as equal objects described by “has aches”.

2.2.4 Semantic Analysis

The step after constituency parsing is semantic analysis. This method analyzes the context and grammar in the surrounding text to correctly identify the proper meaning of more than one definition of words. Basically, by learning all the words in content, semantic analysis accepts natural language material as an input to understand the text's actual meaning. This distinguishes the components of the content from different logical and grammatical roles (Evangelopoulos, 2013).

2.2.4.1 Rapid Automatic Keyword Extraction

Rapid Automatic Keyword Extraction (RAKE) is a widely used NLP technique for semantic analysis. It automatically extracts keywords from sentences based on each document without any other content (Zhao, Bai and Zhu, 2010). RAKE can identify different kinds of keywords given known patterns, including simple noun phrases, simple verb phrases, noun phrases with coordination conjunction and verb phrases with coordination conjunction.

This algorithm firstly builds a list of phrases that are between stop terms (for example, “his”, “the” or “by”) in different n-gram tokens and then discards the stop terms. After that, by counting the frequency of each term that occurs in all the phrases, the algorithm calculates each term’s frequency score. Then, for each term, the algorithm adds up the number of times it appears with every other term in all the phrases to produce a ranking co-occurrence score.

To generate the final ranking list of phrases, this algorithm produces a score for each term in the phrase by dividing the co-occurrence score of the term by the term’s frequency score, and then sums the score of all terms in each phrase to calculate a phrase score.

This score implies the importance of a phrase. The theory is that if a term frequently appears in the text but distributes widely through phrases, it is a less important term and decreases the rank of phrases in which it exists. On the other hand, if the term appears continuously in the matched phrase and has a smaller number of overall occurrences, it will raise the ranking of the phrase in which it is contained.

Although RAKE is a well-known and efficient NLP methodology, its practical application depends a great deal on considerations such as the language of material, the context and the quality of the reports (ZHU and SUN, 2013).

2.2.5 Discourse integration

Discourse integration is a general technique to understand a sentence's meaning based on its surrounding contents. A widely used method to achieve it is Entity Identification, which attempts to identify and classify identified entity mentions in unstructured text into pre-defined categories (Kivenko, 2018). For the QAS data, these categories can be personal names, positions, patient's conditions, accident situations, clinical codes, time statements, other objects, etc.

2.2.6 Pragmatic Analysis

The pragmatic analysis deals with information outside the dataset, which implies knowledge not involved in any input. The pragmatic analysis relies on what terms or phrases have been defined in the real world. In summary, it is the step in which we combine the knowledge gained from the data and the information we acquired from the real word (Barber, 2000).

2.3 Vector Representation

In the domain of picture and speech (by some distance or pixel matrices), we can verify whether the signals or images are comparable. However, natural language is abstract and requires a tool for expressing high-level thinking information (Ramm, 2015).

Text is symbolic data. It is very difficult to identify the relationship between words that have comparable similarity, for example, "Doctor" and "Surgeon" (semantic gap phenomenon). When verifying that the two words are comparable, more background information is required to form an answer. Therefore, it is hard and crucial to converting words into sensible vectors. The main challenge that machine learning algorithms address is how to represent a language phrase mathematically

efficiently. According to the present progress, we have two techniques of representing word to vector. They are the one-hot representation and the distributed representation (Wensen & Zewen, 2016).

2.3.1 The One-Hot Representation

The one-hot representation is one of the most convincing and widely used methods of word representation so far. This technique reflects every word as a very lengthy vector. This vector's dimension number is the vocabulary size of all the archives. Most of this vector's components are 0, and only one dimension has a value of 1, which represents the present term. For instance,
"Doctor" is expressed as [0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 ...]; "Surgeon" is represented as [0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 ...].

Each term in a long list is a 1. If stored sparsely, this One-hot Representation is very unambiguous and concise: that is, assign a numeric ID to each word. For example, the "doctor" is recorded as 1, and the "surgeon" is recorded as 9 (assuming starting from 0) in the previous example. Using the hash table, we can programmatically enforce it to assign a number to each word.

Such a straightforward representation technique with the Maximum Entropy, SVM (Support Vector Machine), CRF (Conditional Random Fields) or other algorithms can effectively complete multiple mainstream assignments in the NLP field (Cilliers, 1992). However, it is not appropriate to apply this technique in this project. As
1. The vector dimension increases as the number of words in the sentence increases;
2. It is impossible to convey relevant information between words at the linguistic level as words are separated expressed;
3. The vector cannot help to classify words, which violates my original intention.

2.3.2 Distributed Representation.

The traditional one-hot representation merely symbolizes the term and does not contain any semantic details. How to integrate semantics into displays of words? Harris suggested that a distribution hypothesis from 1954 offers the theoretical foundation for this concept (Harris, 1954): words with comparable contexts have comparable semantics. Firth further elaborated and clarified the distribution hypothesis in 1957: the semantics of a word is characterized by the words around it (Firth,1957).

Until now, distributed word representation techniques can be split into three classifications according to distinct modeling: distributed representation based on the matrix, distributed representation based on clusters, and distributed representation based on neural networks (Waskom & Wagner, 2016). Although these distinct methods of distributed representation use distinct tools to acquire word depictions (since these methods are all based on the distribution hypothesis). Their key concepts are also composed of two components: first, choose a way to describe the context; second, choose one model to depict the connection between a word and its context.

Matrix-based distributed representations are often referred to as a distribution semantic model. Under this expression, a line in a matrix represents the target word, which defines the distribution of the word context. Since the distribution hypothesis considers words with comparable situations and their semantics are comparable, under this depiction, the semantic similarity of two words can be converted into the spatial distance of two vectors.

Since the distributed representation not only represents the distance between words, but also represents a significant level of the ideal surrounded content for the word. A practical property for the word vector is that they can be summed together. For

similar sentences, by adding each word of each sentence together, the summed vectors are similar (have short distances to each other). This provides the possibility to apply numerical classification methods to text content data.

2.3.2.1 GLOVE Model-Based Word Vector

The widely used Global Vector model (GLOVE model) is a typical Matrix-based distributed representation. It is a technique of decomposing a "word-word" matrix to obtain a word representation (Chen et al., 2017). Another distributed representation method is the neural network-based distributed representation. However, using an N-Grams with word order data as the context to compute the similarity is technically impossible for this project. The context of the target words is the words around the original word. N-Grams should be used as we should consider all the possibilities of the context. However, as N rises, the total number of N-Grams rises exponentially. Then we will encounter the dimensionality and time issues when using this method to represent a considerable number of words.

2.3.2.2 Neural Network-Based Word Vector

The neural network-based distributed representation is the same as other distribution representation methods. Based on the distribution hypothesis, the focus is on the background representation and the connection between context and target words.

A neural network-based distributed representation is generally converted using Neural Network Language Model (NNLM). By representing the word vector using NNLM, the representation of the text is transformed into continuous dense data like images and languages, so that the deep learning algorithm can be migrated to the text field (A New Computational Model of Language Development and Language Processing, 2012). The images below demonstrate the two models: Continuous Words Bags (CBOW) and Skip-gram in the word vector post of Google. The concept is

to convert text information matrix from a high latitude sparse neural network into continuous dense data comparable to pictures so that the deep learning algorithms can be migrated to the domain of the text.

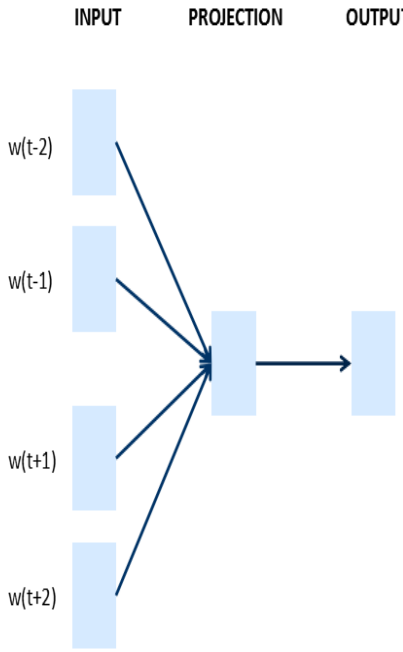

Figure 2. CBOW Structure Diagram
(Using context to predict words)

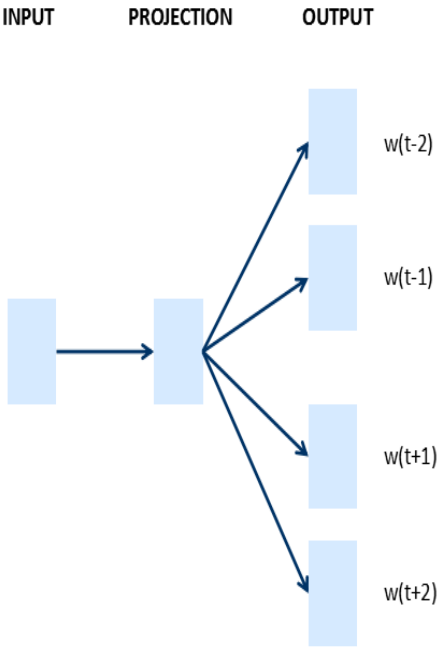

Figure3. Skip-Gram Structure Diagram
(Using words to predict context)

The NNLM can obtain the word vector when learning a language model. The neural network word vector representation, through neural network methods, determines the background and connection between the context and target words. The most significant benefit of techniques with the flexibility of neural networks is that they can depict a complicated context. When the neural network represents the N-Grams, the N-words can be combined, and the number of parameters grows only at a linear rate. The neural network model can fit a more complicated context and contains more considerable semantic information in the word vector.

2.4 Text Classification

Text classification, also known as text tagging, basically categorizes text into organized groups. By using Natural Language Processing (NLP), text classifiers can analyze text automatically and then allocate a series of predefined labels or classes based on the text semantic meaning. Text classification is the goal of many NLP projects and the core technology of large-scale unstructured text file management systems (Boroš and Maršík, 2012).

Unstructured text is everywhere, including messages, online chats, blogs, and medical records but it is difficult to extract meaning from this information unless it is arranged in some form. This used to be a difficult and expensive task as it takes time and resources to manually organize the information and build handcrafted guidelines that are hard to maintain. Text classifiers with NLP have proven to be a great alternative that is cost-effective and easy to implement for scalable structuring textual information. Text identification is becoming an increasingly important part of different fields as it allows information to be easily obtained from large volume of text and complicated processes to be streamlined.

In variety of fields, text classification techniques have been applied maturely. For example, mail technology utilizes text classification to decide whether incoming mail is spam or not. Forums use text classification to decide if posts should be marked as offensive. However, in the medical field, this technology has not been applied widely. As for the classification of unstructured medical text documents, including patient files, medication records and medicine instructions, the accuracy of text classification still has much room for improvement.

These are two instances of text classification, which both categorize a text document into one of a collection of predefined topics. The first classification is based primarily on keywords in the text to address topic identification problems. Another common

type of text classification is an evaluation of semantic content, which might be based on the text's polarity or text's in-depth opinions. For both classification procedures, an accurate classifier is needed. Generally, in a machine learning methodology, it is proposed that different models should be evaluated independently regardless of their conceptual performance since their accuracy depends on the training data set. A few algorithms are generally preferred in the text classification field (SVM, Naive Bayes, multinomial regressions, and random forest). Since there are a considerable number of alternative well-performed classification methods, considering both accuracy, time, the computing power needed and interpretability, Random Forest and SVM were used and introduced here.

2.4.1 Random Forest

A Random forest is an ensemble classifier used in machine learning by using multiple independent decision trees to add a class tag to each output record.

Each tree in the random forest is trained by the input data and generates a class prediction. The class with the most votes becomes the final prediction. Such large numbers of uncorrelated trees, which act as a group, will outperform any of the individual component models (Classification: Practice - Random Forest, 2018).

By using multiple classifiers and their diversifying results, the reliability of the approach is increased and is unlikely to overfit in contrast with single classifiers.
The classifier trees can be trained using bagging or boosting. For bagging, the decision trees are trained for a considerable number of times on a randomly selected subset of the training data. This approach showed that the result in the training set was less sensitive to noise.

For boosting, iterative training of the trees is done on the entirety of the training data. The iterative learning aims are to learn weak classifiers considering distribution

and apply them to a strong final classifier to adjust the weight of previous misclassified records. Thus, this method is more precise than bagging, and classification variance and bias were reduced.

The number of generated trees and variables should be identified before the training process. The number of trees determines the complexity of the model and the splitting condition of the trees is based on the variables

The random forest has become a promising method for text categorization due to its algorithmic simplicity and prominent classification performance for high-dimensional data. One of the most common forest construction methods is to randomly select a subset of features at each node to grow branches of a decision tree, and repeat the process using the bagging method until it comes out with a forest.

Throughout this phase of forest growth, topic-related and insightful features would have a great opportunity to be overlooked. Consequently, weak trees will be formed from these subspaces, the average discriminative capacity of these trees will be decreased and the random forest error boundary will be expanded. Therefore, when a large proportion of such weak trees are generated in a random forest, the forest is likely to make a misguided judgment. This project purposed to use all the features to grow trees and to use the boosting method to adjust the misclassified result for each time new trees are generated. By doing this, the performance of the final random forest classifier can be improved (Vens & Costa, 2011).

2.4.2 T-SNE

T-SNE (t-distributed stochastic neighbor embedding) is an unsupervised dimensionality reduction machine learning method, which converts high dimensionality data to low user-defined dimensionality data. Different from other dimensionality reduction methods (for example, Principal Component Analysis and

Multidimensional Scaling), T-SNE neither extract information from the original dataset, nor calculates the correlation between variables (Common Text Mining Visualizations, 2017). It only focuses on preserving the similarities between records in their original high dimensionality space. Therefore, T-SNE is the best choice for dimensionality reduction before using discriminant functions to classify the high dimensional data.

T-SNE strengthens SNE which has trouble scaling and fails to distinguish the distance from point clusters to single points nearby. In T-SNE, the high-dimensional Euclidean distances between data points are transformed to a matrix of similarities which is defined as the conditional probability of xi as neighbor of xj. Neighbors are chosen for their probability density under a Gaussian distribution centered at xi. Then, T-SNE uses a t-distribution to measure similarities in a low-dimensional space to minimize variations (represent as Kullback-Leibler's divergence) between these similarities using Gradient Descent.

Since the project aims to visualize the word vectors and classify the document vectors, T-SNE is suitable for this task not only because of its efficiency and scalability. By breaking down high definition information into two dimensions, T-SNE can maintain similarities of data points strongly utilizing likelihood distributions both of its initial dimensionality and decomposed dimensionality, which can provide an ideal result for data visualization to verify whether the word vector model is trained well. By checking whether similar words are clustered together in a two-dimensional graph, the performance of the word vector model can be easily determined. As the word vector model is well-trained, the document vector can accurately represent the documents. And for the document vector, T-SNE can also be used to visualize documents as a scatter plot where similar documents will be clustered together theoretically.

2.4.3 SVM

Support vector machine (SVM) is a supervised learning algorithm used for classification based on discriminant functions. SVM is used in this project to construct hyperplanes on the document vector matrix to classify reports (B. Chrystal & Joseph, 2015). These constructed hyperplanes represent the greatest margins between the groups. When data is not linearly separable in its dimensions, a kernel trick is used to map the data into a higher-dimensional space in which a linear separation is possible. The support vectors are found analytically as a convex optimization problem. Once the training is complete, SVM requires only the support vectors for predictions. When there are multiple classes, strategies such as one-versus-the-rest, pairwise classification, and the multi-classification formulation can be used. For one-versus-the-rest, the most frequently used multi-classification algorithm, each support vector distinguishes each class from the rest.

2.5 Text Mining and NLP in R

2.5.1 R

The proposal aims to use R to analyze this text data. R is a free programming language and programming environment for statistical analysis and graphics supported by the Foundation for Statistical Computing. Research language (R) is widely used among statisticians and data miners to develop statistical software and data analysis. R is highly extended with user-provided packages for specific functions or specific areas of study. Because of its legacy in S, R has more object-oriented software facilities than most statistical computing languages. R extension is also mitigated by lexical definition rules.

2.5.2 R General Analytics Packages

R provides different tools to work with data frames for data manipulation. Using these packages, users do not need to write complicated code to achieve loop and discriminant tasks. These packages can greatly save analysts' time and energy by using a large amount of built-in functions, which is also why R is more friendly to different classes of users compared to other languages. The basic packages used in this project include "dplyr", "widyr", "plyr", "igraph", "ggplot2", "ggraph", "igraph", "compare" and "xlsx". These packages were mainly used for data cleaning, data frame transformations and visualizations.

2.5.3 R Text Mining Packages

There are a considerable number of packages for text processing in R. Some of them are built to convert messy text data to a tidy format, and some of them are built for specific text analytics functionalities (for example, tokenization, stemming, and calculating the word correlation). For the text mining purpose, the project mainly used "tidytext", "tidyverse", "janeaustenr", "topicmodel", "broom".

2.5.4 R NLP Packages

General natural language processing methods are generally applied based on a java-written tookit called Apache Opennlp, which supports tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, language detection and coreference resolution. "Rjava" and "opennlp" packages in R can provide an environment to run this java-based package. In addition, other NLP packages including "nlp","udpipe", and "opennlp.model.en" were also used for constituency parsing and keyword identification. For word vector generation, "text2vec" and "keras" were used.

2.5.5 R Machine Learning Packages

The “Caret” package in R provides a wide range of choices to change models. It can be used to train classifiers using different methods including SVMs, decision trees and random forests. Thus, for the model training part, this project mainly used ”caret” to build models and used ”tictoc” to manage time and space cost. Other packages including ”tm”, “randomforest” and ”e1071” were used to better modify the parameters of the models.

3. Methods

In the previous section, the theoretical knowledge of the techniques used in this project was introduced. This section will focus on how the project was implemented this semester to classify the reports.

3.1 Methods in Semester 1

- First, tokenization, stemming, and removing the stop words were implemented as an essential data cleaning step.
- Second, to explore the data set, the number of words in all the documents were counted and sorted by their frequency. By using the word frequency table generated, a word cloud showing the most frequent 150 words was drawn to visualize and better understand the composition of the words in the QAS data.
- Third, to understand the accident frequency and density, the frequency of the report records for each postcode was counted and ranked from high to low. Thus, the population data from the Australian Bureau of statistics was merged with the original QAS data to calculate the car accident density (Density ≈ Accidents/Population). By merging the accident frequency and density with the postcode, Queensland accident frequency and density can be visualized geographically.

- Fourth, for identifying accident-prone roads and the main accident causes in suburbs, the pairwise correlation between common words (appeared more than 100 times) was calculated and filtered.
- Fifth, other attempts including discovering the usage of “head” in the reports and the transferred hospital of the accidents were also implemented using n-gram tokenization and word correlation respectively.
- As a result, by exploring the word correlation, the main accident causes in certain areas can be identified. In addition, lists of the high frequent accident roads and suburbs were also generated to identify the accident-prone and better understand the data.

3.2 Methods in Semester 2

The classified QAS data was considered (29755 of 91384 cases are cause-classified truck collision reports. 16705 cases after deleting missing causes. These 16705 cases will be referred to the whole dataset in the following.)

- First, the whole dataset was divided randomly into a training set, a test set and a validation set (based on the proportion 6:2:2). A lexical analysis of training data was conducted. The training data was tokenized into a one-word-each-term format. After that, this tokenization data was cleaned by removing stop words and stemming process. Thus, a TF-IDF matrix was generated to represent the composition and distribution of the words for each of the reports.
- Second, the NLP constituency parsing was applied to the whole dataset to tag the part of speech for each word. For the generated constituency parsing trees (which identify the part of speech and sentence structure in each sentence), rapid automatic keyword extraction was used to discover the simple noun phrases and simple verb phrases given certain patterns respectively. Using the identified keywords (simple noun phrases and simple verb phrases), two keyword TF-IDF matrices were generated to represent a keywords composition and distribution for the training set and test set respectively.

- Third, two random forest models were built using training TF-IDF and training keyword TF-IDF respectively to classify reports based on the cause of the accident (Because only causes were classified in the report and there is no other classified information). After that, the test TF-IDF and test keyword TF-IDF were used to evaluate the performance of these two models.
- Fourth, a word-vector representation matrix was generated using the Glove based word2vec technique using the whole dataset. By adding all the word vectors in each document, a document vector matrix was generated which was used to compute the similarity between documents based on the vectors' distances. After that, a dimensionality reduction method (T-SNE) was used to visualize the document vector matrix on a two-dimensional plot.
- Fifth, An SVM model was built based on the selected training data from the T-SNE result to draw margins between document vectors based on the document vector matrix. The test data and validation data were used to evaluate the performance of the SVM model.
- As a result, by comparing the performance of different models, the analyzing strategies for the next step for QAS data was determined. Ultimately, most of the information in the reports can be converted into structured data for better analysis.

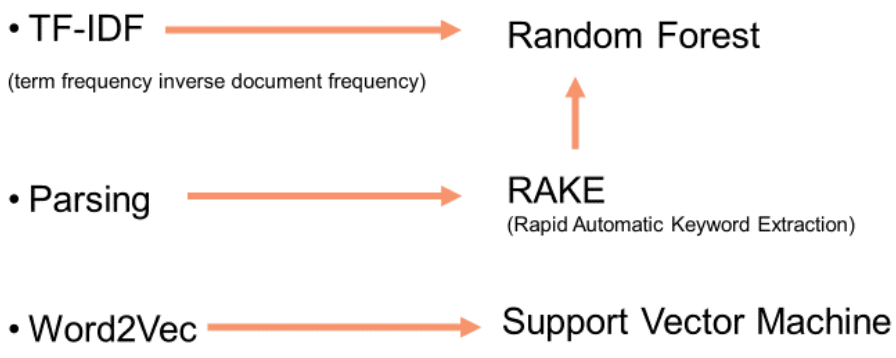

Figure 4. Illustration of Workflows in Semester 2

4. Results

4.1 Data Exploration and Accident-Prone Identification (Results for Semester 1)

After tokenization and data cleaning, the number of words was counted and sorted by their frequency. There are 59,999 words in the dataset, and the five words appearing at the highest frequency are pt (which means patient), nil, pain, vehicle, car. The events that most ambulance reports respond to can be referred based on the word frequency table. Most of the patients that have no significant problems (nil). A word cloud was drawn for a general visualization. The word cloud below shows the 150 most common words that appeared in the dataset. The more common the words appear in the middle area. The most talked-about human anatomy parts in the QAS report are the head, spine, chest, and neck. In addition, the words about the patient's condition also appear in the word cloud, for example, pain, trauma, nausea, and tenderness. Most of these words describe minor discomfort. It can be referred that most reports do not involve serious accidents. Interestingly, we can see from the word cloud that security devices such as seatbelt, helmet, and belt are also often mentioned.

	word	n
1	pt	343785
2	nil	131420
3	pain	107961
4	vehicle	78270
5	car	62007
6	qas	61100
7	stated	36077
8	rtc	35276
9	head	32565
10	gcs	28613
11	15	26782
12	denies	26395
13	front	25944
14	chest	25442
15	driver	25114
16	extricate	24293
17	left	24176
18	road	23884
19	neck	23694
20	scene	23590
21	approx	23351
22	injuries	22324
23	damage	21454
24	patient	21312
25	spine	21150

Table 2. The Most Frequent 25 Words (Generated by R)

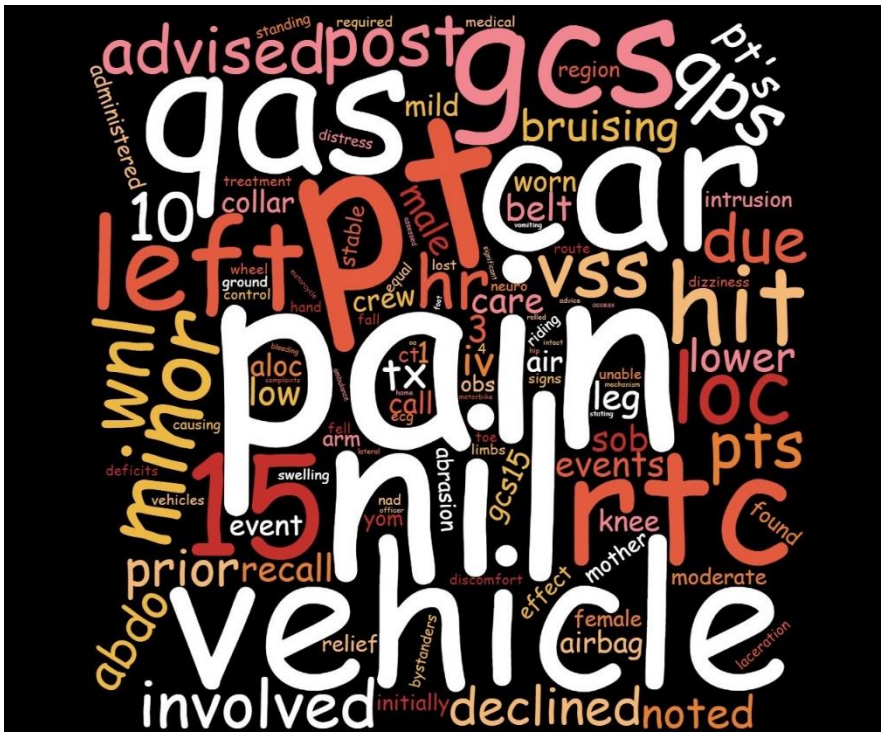

Figure 5. The Most Frequent 100 Words Word Cloud (Generated by Tableau)

4.1.1 Accident Frequency and Density

To understand the accident distribution, the frequency of postcode where the accidents happened was counted and ranked from high to low. The graph below shows the first 16 records of the accident frequency table. For 4350 (Toowoomba), 4740 (Mackay) and 4670 (Bundaberg), there are more than 1000 accidents that were serious enough to call an ambulance in the past four years.

postcode	Frequency
4350	1546
4740	1215
4670	1119
4570	942
4207	929
4211	921
4305	814
4870	810
4306	767
4510	711
4114	623
4215	616
4214	592
4000	587
4503	579
4702	564

Table 3. Accident Frequency by Suburbs in Queensland (Generated by R)

Based on the accident frequency, a Queensland geographical accidental map was visualized. In the accident map (2015-2018) shown below, a deeper color of a specific area means car accidents happened more frequently in that area. As the graph is shown, most of the accidents happen in the eastern coastal areas. Specifically, Brisbane and surrounding suburbs have a higher incidence of accidents compared to other suburbs in Queensland. However, this conclusion has not much practical significance. Because the eastern coastal areas are initially a population gathering place, vehicle accidents will naturally be more.

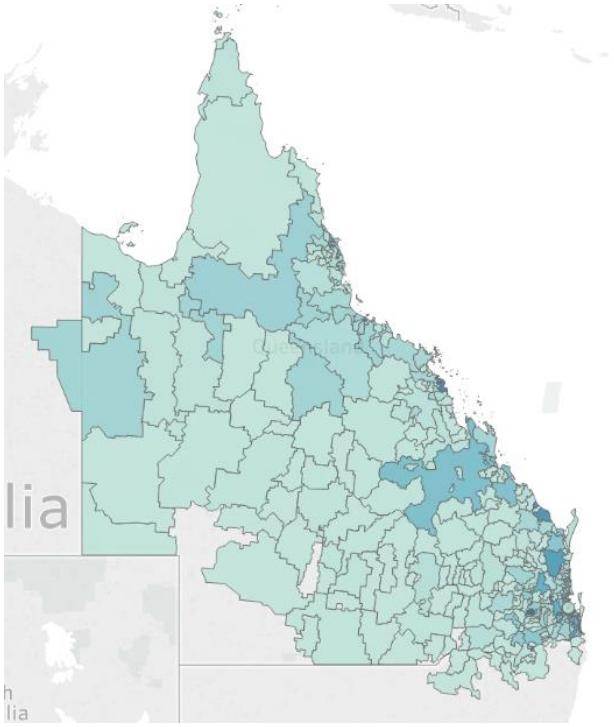

Figure 6. Accident frequency map by suburbs in Queensland (Generated by R)

Thus, using the population data from the Australian Bureau of statistics, the density of accidents can be calculated roughly. The population data divides Queensland into 187 regions based on geographical divisions, as is shown below. These areas encompass all suburbs. As accurate suburb population data is not known, the average of the local population as the estimated value of each suburb in that specific region. I use the accident frequency divided by the estimated population to calculate the accident rate and visualize it on a map.

Rank	Urban Centre	Population				Region
		2016 census		2011 census		
1	Brisbane	2,054,614	[1]	1,874,427	[2]	South East Queensland
2	Gold Coast–Tweed Heads (Gold Coast part)	540,559	[3]	478,107	[4]	South East Queensland and Northern Rivers
3	Sunshine Coast	243,337	[5]	209,263	[6]	South East Queensland
4	Townsville	168,729	[7]	157,748	[8]	North Queensland
5	Cairns	144,730	[9]	133,893	[10]	Far North Queensland
6	Toowoomba	100,032	[11]	96,567	[12]	Darling Downs
7	Mackay	75,710	[13]	74,219	[14]	Central Queensland
8	Rockhampton	61,214	[15]	61,724	[16]	Central Queensland
9	Hervey Bay	52,073	[17]	48,680	[18]	Wide Bay-Burnett
10	Bundaberg	50,148	[19]	49,750	[20]	Wide Bay-Burnett
11	Gladstone	33,418	[21]	32,073	[22]	Central Queensland
12	Maryborough	22,206	[23]	21,777	[24]	Wide Bay-Burnett
13	Mount Isa	18,342	[25]	20,570	[26]	Gulf Country
14	Gympie	18,267	[27]	17,285	[28]	Wide Bay-Burnett

Table 4. Queensland Region population (Gain from Australian Bureau Statistics)

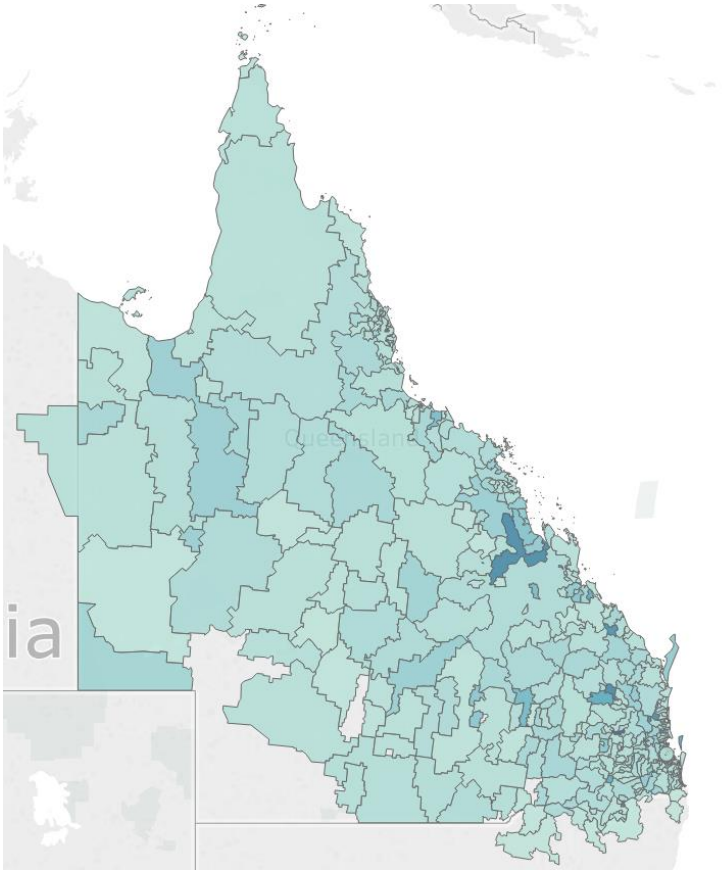

Figure 7. Accident Density Map by Suburbs in Queensland (Generated by Tableau)

This approach considers the impact of the number of residents in each region on road traffic. However, the geographical resident population may not fully represent the degree of regional congestion. For example, in 4008 Peugeot, the accident rate is considerably high. Because there are not many people living there, the passenger flow is still large given that Brisbane airport is located there. Large passenger flow might increase car accidents risk by exceeding the road capacity, but based on the probability theory, car accidents would undoubtedly increase if there are more cars on the road. Unfortunately, so far, the analysis cannot tell whether accidents happened because of the original road design problem or just a larger cardinality because there is no passenger flow data for every road. However, the accident rate calculated using this method can reflect the severity of car accidents to a certain extent in most of the areas.

In the accident rate map, the darker the color of an area, the higher the accident rate there. Compared to the accident rate map, the color of the east coast is lighter, especially for the Brisbane and Gold Coast region. It indicates Brisbane and Gold Coast's good road condition and driver awareness as well.

However, the accident rate in some areas is still comparatively high. For example, after eliminating the transportation hub, the top three suburbs with the highest accident rate are 4402 (Cooyar) with a 23.3% accident rate, 4612 (Hivesville) with 17.5%, and 4705 (Mount Gardiner) with 17.3%. It is a high density of accidents, indicating that for every five residents in these areas, for every four years, about one of them might experience a car accident.

Although the passenger flow was not taken into consideration, the calculated accident rate can still demonstrate some problems. A high accident rate in these areas may be due to a flaw in the road design, or natural reasons or insufficient awareness of the overall traffic safety of the residents. Any high accident should be given attention. Therefore, I extracted the suburbs with an accident rate greater than 5% and did further analysis to identify the cause of the accident. There are more

than 32 of this kind of area in Queensland.

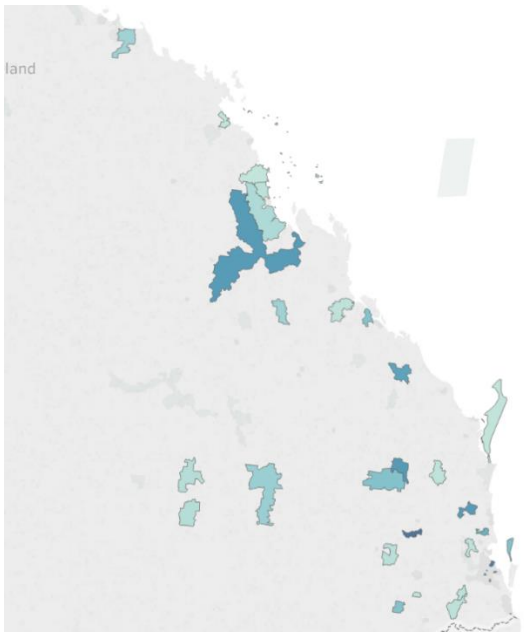

Figure 8. Accident Prone Suburbs in Queensland (Generated by Tableau)

4.1.2 Accident-Prone Roads and Transferred Hospitals

A pairwise correlation was used to calculate the word correlations of all the words that appear in the report. Since the QAS data set is large and covers most of the commonly used words in natural language, it is not possible to determine the causes of accidents in each region from the word correlation. For example, a word may have a higher correlation with a postcode number, but it cannot be significantly higher than that with the others unless it is a specific word.

However, accident-prone sections in various regions can still be identified. The reason is that the words used to name road segments are unique (mostly by name); they tend to be used less in natural language. Similarly, the hospital for each ambulance that received patients can also be identified using this method. The QAS report does record the transferred hospital name using a specific abbreviation format, which is also different from the commonly- used words. It is worth noting that, only

when the accident is comparatively severe, will the ambulance transfer patients to the hospital.

Thus, a higher hospital-suburb pairwise correlation means the corresponding area was more likely to cause more severe accidents. Of course, this cannot be used as an accurate measure because if an accident occurs between two hospitals, the patients transported by ambulance may be spread across two different hospitals, which dilutes the hospital-suburb correlation. However, since this is not the case, this pairwise correlation can still indicate the severity of the accidents in the area to some extent.

Therefore, firstly, the correlation was calculated between all words that have appeared more than 100 times in the collection. Secondly, by filtering out the results, one of the words to be a number and greater than 1000, the outcome is the correlation between postcode and other words. Thirdly, for identifying accident-prone roads, I anti- joined the result with the postcode-suburb table to ensure that the correlated word pairs are not the postcode with the suburb itself; Moreover, for identifying transferred hospitals, I filtered words from the pairwise correlation results ending with "h" and not in the daily used dictionary as the hospital-suburb word pairs . Finally, I chose the correlation with a correlation greater than 0.1 to display. This method has been able to derive the correlation of the suburb and hospital and incident sections to a large extent.

For each pair of the results, more than 10% of the accidents in the region represented by the postcode occurred on their corresponding streets or more than 10% of the patients suffered from accidents in the region represented have transferred to the corresponding hospitals. Since this is calculated using correlation, the result will be slightly larger than the actual percentage.

	Postcode	Word	correlation
1	4575	nicklin	0.611489
2	4511	bribie	0.472769
3	4865	gillies	0.415297
4	4881	kennedy	0.408034
5	4564	david	0.342407
6	4069	moggill	0.340281
7	4173	wynnum	0.309445
8	4170	wynnum	0.305427
9	4022	deception	0.285334
10	4068	moggill	0.284096
11	4702	capricorn	0.260987
12	4280	lindesay	0.259688
13	4066	milton	0.245829
14	4021	anzac	0.238344
15	4074	centenary	0.235537
16	4870	mulgrave	0.235301
17	4701	yeppoon	0.234002
18	4816	flinders	0.232755
19	4300	redbank	0.232744
20	4820	gregory	0.231657

Table 5. High Correlation Pairs of Postcode and Street Fragment (Generated by R)

For postcode-street correlation, for example, more than 61% of accidents that happened in 4575 Sunshine Coast were on Nicklin Way, and more than 47% of accidents that happened in 4511 Ningi were on Bribie Island Road.

In a considerable number of areas, traffic accidents occurred in a very concentrated region and were all in the same area. These cases may have certain similarities and correlations. Traffic accident-prone regions can be identified even without reading all the reports manually. Based on the correlation result, we can point out all the black spots even without analyzing the reasons for this. We can also know that there are often some traffic hazards in these sections so that there will be accidents in the vicinity. Some measures can be implemented for these concentrated traffic accidents.

	postcode	locality	correlation
1	4825	mibh	0.58471566
2	4870	cbh	0.400646475
3	4850	idh	0.33148459
4	4807	adh	0.318770201
5	4305	igh	0.293999561
6	4814	tth	0.292200711
7	4701	rh	0.254606787
8	4700	rh	0.252330677
9	4810	tth	0.240870869
11	4818	tth	0.230401066
12	4306	igh	0.227525143
13	4815	tth	0.217351445
14	4860	idh	0.213242013
15	4865	cbh	0.206990261

Table 6. High Correlation Pairs of Postcode and Hospital Fragment (Generated by R)

For postcode-street correlation, for example, more than 58% of the accidents that occurred in 4875 Mount Isa were severe enough to send patients to the hospital. More than 58% of patients in 4825 Mount Isa have been transferred to MIBH (Mount Isa Base Hospital). More than 40% of patients in 4870 Cairns have been transferred to CBH (Cairns Base Hospital). Given such a table, we know where the locations are more prone to severe accidents. These places may require more medical facilities

After considering the accident-prone suburbs with higher than 5% accident rate found in the previous section, word correlation was calculated again specifically for these areas. Because using a smaller data set can more accurately estimate the proportion of accidents happened in a street across the whole suburb. In this way, it may be possible to find out why the traffic accident rate in these areas is so high or where car accidents happened explicitly in these areas to prevent accidents.

Postcode	population	locality	correlation
4155	1079	cleveland	0.79952672
4600	959	wide	0.578596058
4117	1211	wembley	0.497419185
4106	1256	granard	0.374876217
4738	1100	sarina	0.349133992
4738	1100	bruce	0.345379788
4117	1211	logan	0.338603456
4106	1256	ipswich	0.330238651
4738	1100	highway	0.250339601
4738	1100	south	0.238106797
4106	1256	forklift	0.215635738
4738	1100	mackay	0.188108132
4738	1100	ambulatory	0.174851961
4600	959	responders	0.171248486

Table 7. High Correlation Pair of Postcode and Street in Accident-Prone Suburbs Fragment (Generated by R)

After reducing the amount of data, it is not very meaningful to explore small suburbs with only a few roads. Because there are only a few roads in these areas, the road names in the area and the word correlation of the postcode will naturally be high. Therefore, removing areas with estimated populations greater than 500 from the dataset will ensure that our results contain a certain number of roads. As shown in Table 7, for the 4155 Chandler area, more than 79% of the accidents occurred on Old Cleveland Road. There are many other similar car accident-prone roads in other suburbs. These have been attached to the list.

4.1.3 Other Explorations on N-Grams

As mentioned above, some words have a very high word frequency, such as head and belt. To see how these words are used in the report, terms were tokenized in a 5-grams form and filtered the word head and belt to display. As can be seen in Table 8 below, when the report includes “belt,” it usually comes with wearing a seat belt. When the report mentions “head,” it has mostly been “head to toe examination” or “not hit the head.”

The word “head” has appeared 18915 times while the word “belt” has appeared 8778 times in 5-grams. Then a sentiment analysis of the words around the target ”head” and “belt” was implemented. It turns out that head appeared 5804 times in a positive way and 13111 times in a negative way, while belt appeared 2098 times in a positive way and 6680 times in a negative way. The result is quite interesting and worthy of further exploration.

	word1	word2	word3	word4	n
1	wearing	a	seat	belt	854
2	was	wearing	seat	belt	494
3	seat	belt	worn	nil	386
4	deployed	seat	belt	worn	305
5	seat	belt	worn	pt	303
6	pt	wearing	seat	belt	266
7	a	seat	belt	and	233
8	wearing	her	seat	belt	194
9	wearing	seat	belt	nil	181
10	seat	belt	worn	and	179
11	wearing	seat	belt	and	177
12	wearing	his	seat	belt	172

Table 8. Usage of the word “belt” (Generated by R)

	word1	word2	word3	word4	n
1	on	head	to	toe	1114
2	did	not	hit	head	1057
3	pt	denies	hitting	head	711
4	head	to	toe	revealed	511
5	not	hit	his	head	485
6	not	hit	her	head	407
7	head	to	toe	pt	403
8	head	to	toe	nil	367
9	pt	denied	hitting	head	343
10	head	to	toe	assessment	337
11	head	to	toe	examination	313
12	not	hit	head	nil	304

Table 9. Usage of the word “head” (Generated by R)

4.2 Topic Modeling

In order to explore the causes of car accidents and the types of car accidents in more depth, the reports according to the content of each report and then conduct regional discussions should be appropriately classified. Therefore, an LDA model was used to classify all reports into multiple types of valid topics. Since both the number of topics in the documents and the number of terms in each topic is not set, a range of parameters (Topics from 4 to 20, and Terms from 7 to 12) were used to build a few LDA models.

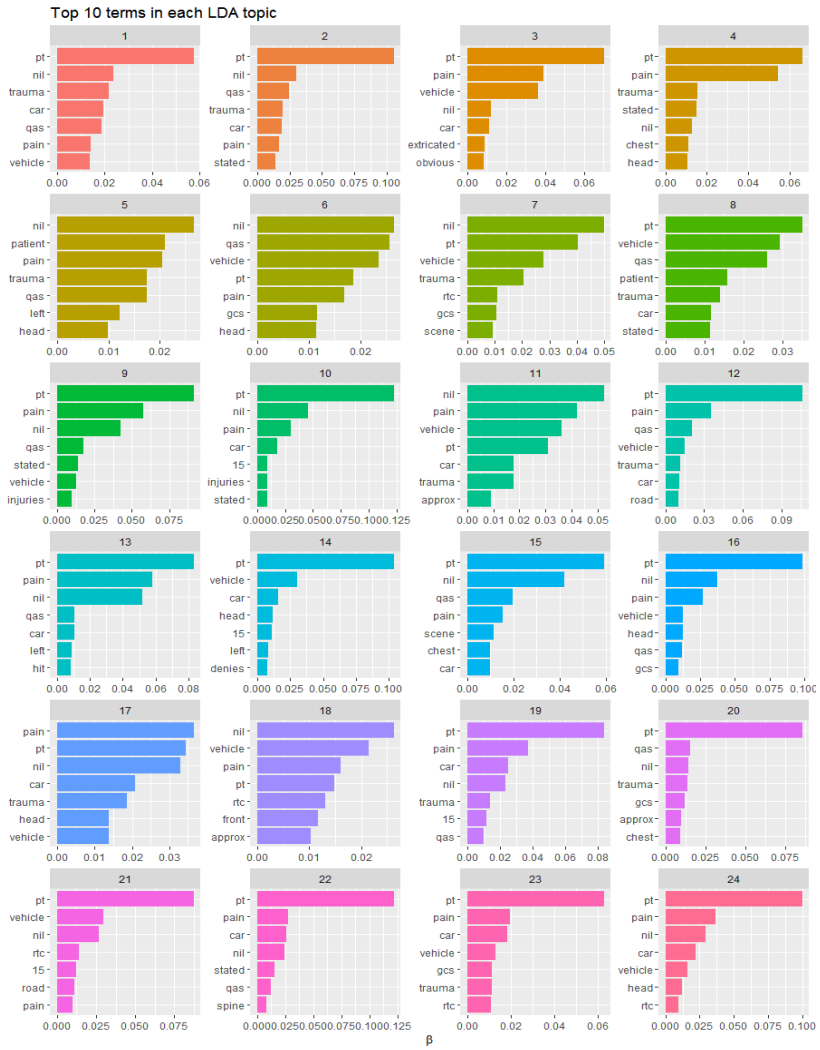

Figure 9. Terms in LDA Topic model (Generated by R)

The fitted models in different parameters perform not effective at all as one of the examples is shown above (parameters used 7 terms and 20 topics). For each topic, there are a lot of overlapping terms, and all these terms do not form a meaningful topic. I have considered taking the frequently appeared words as stop words and removing them before fitting an LDA model.

However, this is not something that can be done in such a general way. Because the words with high frequency, such as pt, nil, and pain, are deleted, there will still be words with high frequency. Words that appear many times cannot be defined as meaningless.

Moreover, each report in the dataset is not independent. QAS sets a set of inspection procedures that apply to a large number of car accidents, which means that in most cases, certain words will be repeated and these words will appear in the LDA model with a high level. A poor fit would be obtained irrespective of whether these words are considered or removed.

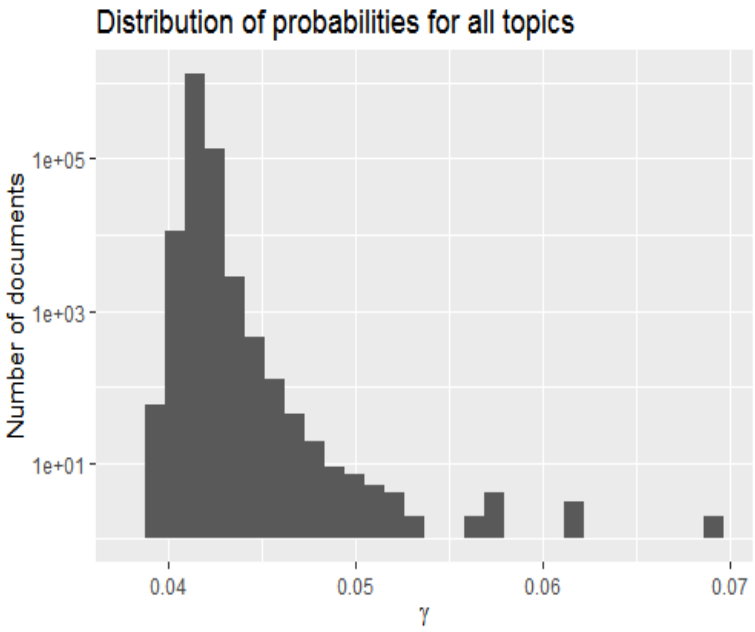

Figure 10. Distribution of Probabilities of All Topics (Generated by R)

So far, the model validation is very inefficient, for both all topics model and each topic model. Most files can only explain 4% to 5% of the topic content. After removing all the common words, the part that the model can explain can even be lower.

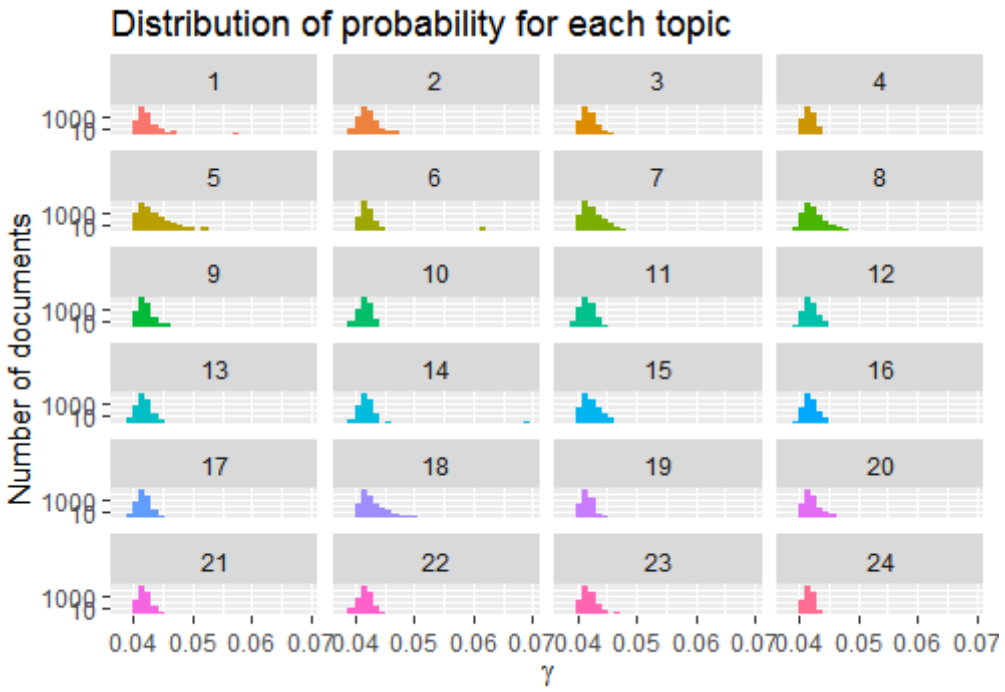

Figure 11. Distribution of Probability for Each Topic (Generated by R)

4.3 TF-IDF Based Modeling

By counting the frequency of each unique word in each report, the Term Frequency-Inverse Document Frequency matrix can be generated. To build an appropriate classifier using this text representation, the cause distribution of the reports needs to be first explored as labels with only few samples in the training data might influence the supervised learning accuracy. Among 16171 records in the training set, 96 % of valid accidents were caused by the top six reasons (14461 cases in 15297 valid records). Thus, only the reports where their incidents cause was in the top six were considered.

4.3.1 TF-IDF and Word Composition

Using the considered reports, the word composition was verified as being different in these reports. Hence the classifier can tell the difference between these reports as the training content is different. Based on the word composition graph, we can see the frequency of words are diverse in different cause reports. The most obviously different cases are non-collision, collision with pedestrian or animals, and collision with stationary objects reports. The non-collision reports mainly focus on bike and helmet, while collision with pedestrian or animal reports and collision with stationary object reports focus on animal, pedestrian, tree, and pole respectively. Since the word composition of other reports is similar, and motor and vehicle collision occupied in most of the cases, the classifier based on this TF-IDF might misclassify reports caused by collision with others, and other unspecified collision to reports caused by motor and vehicle collision.

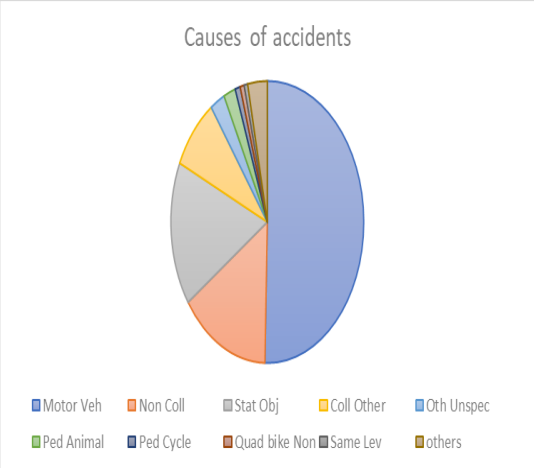

Figure 12. Pie chart of Causes of Accidents (Generated by R)

Subsequently, the whole dataset was randomly divided into a training set (60%), a testing set (20%) and a validation set (20%) for avoiding the overfitting problem. Because TF-IDF reflects the term (word) importance in each document for the whole corpus, to avoid the biases in the evaluation result, TF-IDF matrices were constructed after sampling the whole dataset into the training, test and validation set. Using three individual datasets, three sperate TF-IDF matrices were generated. The training

TF-IDF was used to trained the random forest classifier. The test TF-IDF was used as an input for the model to verify if the classifier works generally and to validate the TF-IDF inspected if the test result was correct.

	id	word	n	tf	idf	tf_idf
1	1	broke	5	0.064935065	5.9618426	0.387132638
2	1	car	5	0.064935065	1.0271888	0.066700574
3	1	forehead	5	0.064935065	4.0052752	0.260082804
4	1	front	2	0.025974026	1.5499538	0.040258541
5	1	glass	2	0.025974026	4.7454473	0.123258371
6	1	hit	17	0.220779221	1.3930771	0.307562480
7	1	injuri	2	0.025974026	1.5454146	0.040140638
8	1	lane	1	0.012987013	3.7757913	0.049036251
9	1	left	5	0.064935065	1.4893470	0.096710847
10	1	merg	1	0.012987013	4.8385377	0.062838152
11	1	nil	2	0.025974026	0.9589035	0.024906585
12	1	passeng	6	0.077922078	1.9663980	0.153225819
13	1	pt	6	0.077922078	0.1421307	0.011075118
14	1	remov	1	0.012987013	2.9965696	0.038916488
15	1	seat	4	0.051948052	1.8604397	0.096646218
16	1	vehicl	9	0.116883117	0.8153661	0.095302531
17	1	window	4	0.051948052	4.0785680	0.211873664

Table 10. TF-IDF Fragment for Training data (Generated by R)

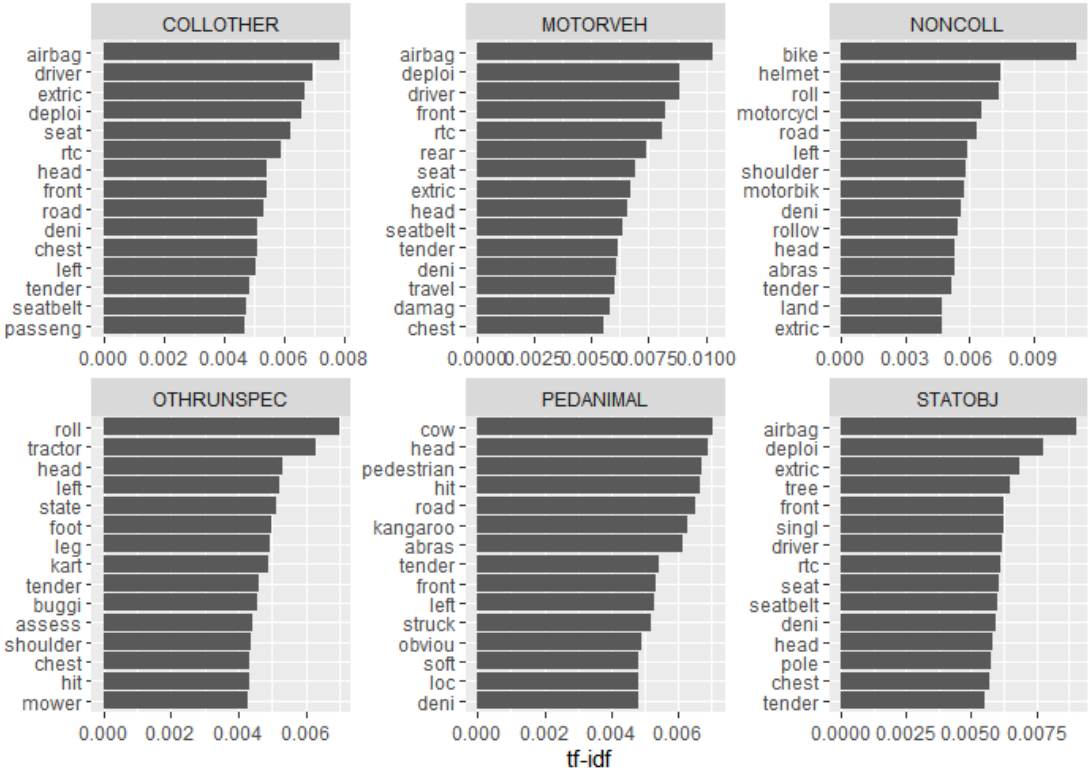

Figure 13. Word Composition for Reports Caused by Top Six Reasons (Generated by R)

4.3.1 Random Forest and Decision Tree

Using the term-frequency inverse document frequency matrix, random forests models were built based on different splitting rules and selected predictors. As shown in Figure 14, the accuracy of the random forest models became stable when using more than 100 predictors (term frequency). As 800 predictors and the extra trees splitting rule were used, the classifier test accuracy was the highest (about 73%). As an ensemble training method, the random forest model consists of multiple trees (200 in this case). Each tree trained using the boosting method is strongly biased and cannot provide a reasonable result given an input. Thus, it is not appropriate to visualize the splitting conditions for a single tree in the forest. However, random forests provide a variable importance ranking, which shows the out-of-bag error of a specific feature. In this project, this variable importance measures that if we remove a certain variable (word) from the training set, how large

the error will increase for the model. Figure 15 shows that “tree”, “single”, “rtc” (road traffic crash), “car”, “pole” are important words to decide the classified outcome. In this case, the output of the classifier will highly depend on if these important words exist and the proportion of them in the input.

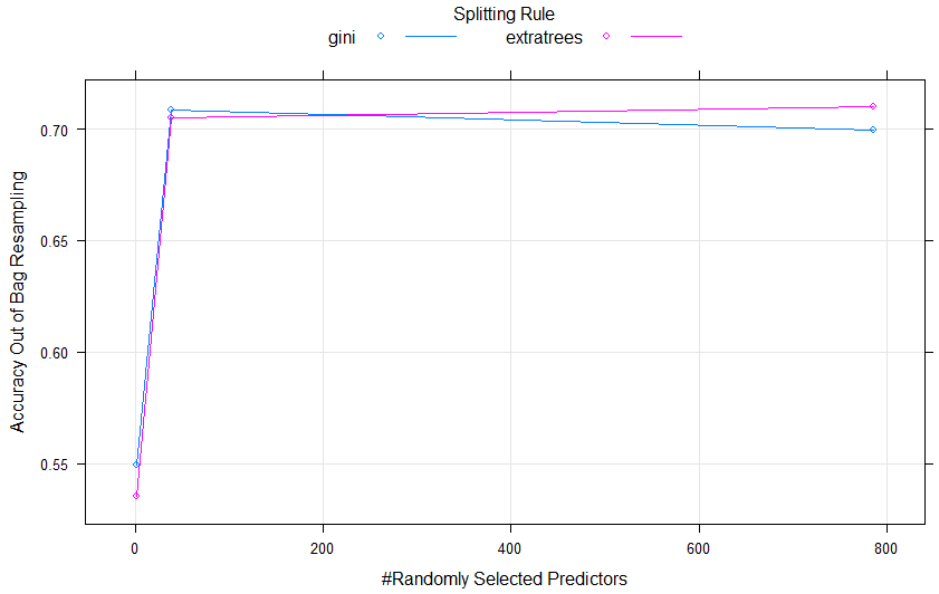

Figure 14. The Random Forest Test Accuracy Using Different Splitting Rules and Parameters (Generated by R)

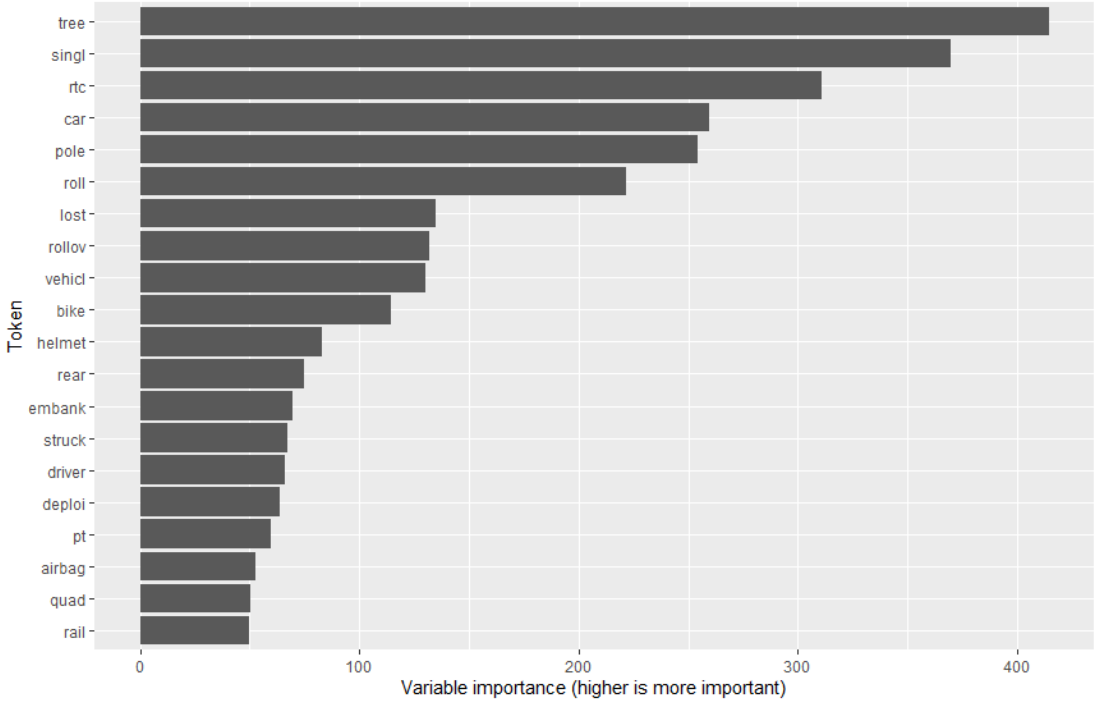

Figure 15. The Random Forest Variable Importance Ranking (Generated by R)

To better visualize how the classifier split based on the conditions, a decision tree model was built as it is the basic element of a random forest. Figure 16 indicates the decision tree splitting conditions based on the TF-IDF value of a word in the input report. For example, when “tree” is considered as an important word or both “single” and “pole” are considered as important words in the report, the case is classified as a stationary object collision. Since this decision tree model was only built for visualization and understanding the random forest model splitting results, the size and depth (complexity) was set to be low. Thus, this model can not predict a certain group of causes (collision with pedestrian or animals, other unspecified collision, and collision with others).

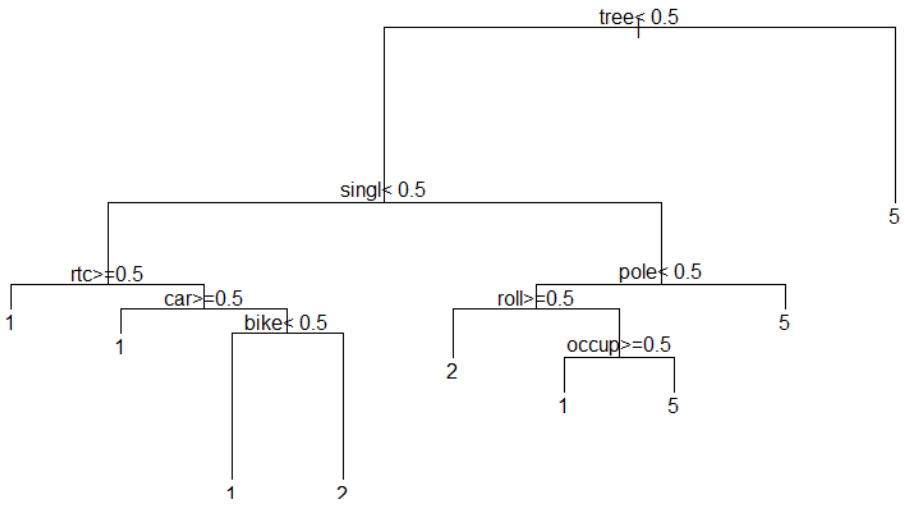

Figure 16. An easy fitted Decision tree for Understanding the Random Forest Model. 1 represents motor vehicle collision; 2 represents non-collision; and 5 represents stationary object collision. (Generated by R)

4.4 Constituency Parsing

Given the final test accuracy of the random forest model (73% as shown in Figure 14), TF-IDF cannot efficiently represent the useful causes information or provides too much redundant information other than accident causes. Thus, the NLP constituency parsing method was used to analyze the POS (part of speech) and sentence structure. As a result, useful information can be extracted and represented as a keyword TF-IDF theocratically. Subsequently, we can build a more practical classifier based on the extracted information. Because the tokenization and stemming turned all the words into their word stems, which might change the original part of speech. It should be noticed that since this constituency parsing step aims to analyze the report's sentence structure, the original report data were used.

4.4.1 Universal Parts of Speech (UPOS)

For each sentence, constituency parsing tagged the parts of speech (POS) of each word. To understand the POS structure in the QAS reports, this project printed the universal part of speech graph as shown in Figure 17. This graph used a tidy abbreviation to represent long content. The full illustrations of the abbreviations can be found in Table 11. In the QAS reports, most of the words are nouns (excluding proper nouns, pronouns and noun phrases with coordination conjunction). The frequency of verb ranked second when not considering punctuations. Other than nouns, verbs, adpositions, and adjectives, other parts of speech occupied a low proportion in the reports.

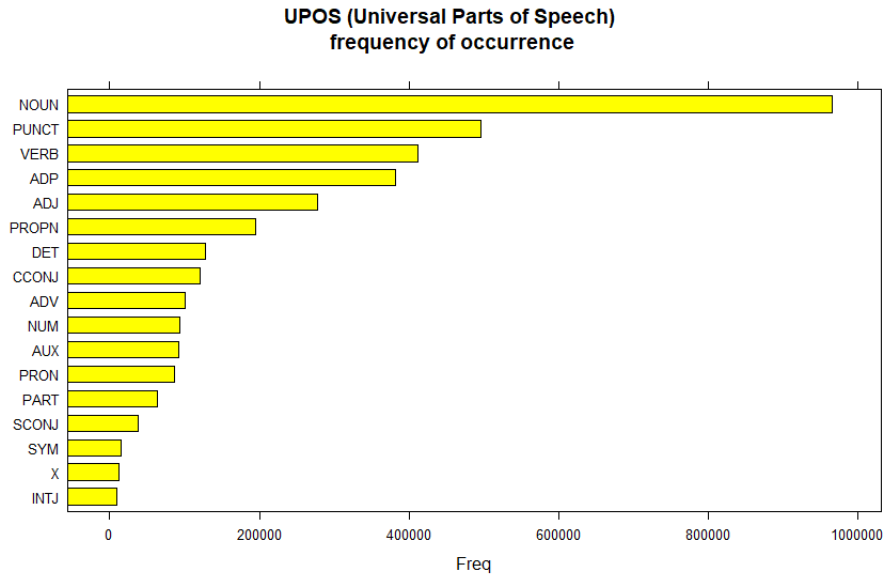

Figure 17. Universal Parts of Speech for all the reports (Generated by R)

ADJ	adjective
ADP	adposition
ADV	adverb
AUX	auxiliary verb
CONJ	coordinating conjunction
DET	determiner
INTJ	interjection
NOUN	noun
NUM	numeral
PART	particle
PRON	pronoun
PROPN	proper noun
PUNCT	punctuation
SCONJ	subordinating conjunction
SYM	symbol
VERB	verb
X	other

Table 11. Illustrations of the Part of Speech Abbreviation (Gained from Wikipedia)

The most frequent occurring nouns, verbs and adjectives were also explored and visualized as shown in Figure 18, 19, and 20. From the graphs, it can be seen that “pt” (“patient”) occurs many times as both nouns, verbs and adjectives. This was because constituency parsing tags the parts of speech not only according to the sentence structure, but also referring to the built-in dictionary where “pt” does not exist. Additionally, as mentioned in the challenges section, most QAS reports were not written in formal English, constituency parsing might not be able to accurately tag the POS of these informal word combinations. However, except the “pt” issue, other parts of speech look fine in the graph. We can refer to some rough information from the graph. Most of the patients have “nil” or “minor” problems after car accidents, they just felt “pain” somewhere (mostly heads).

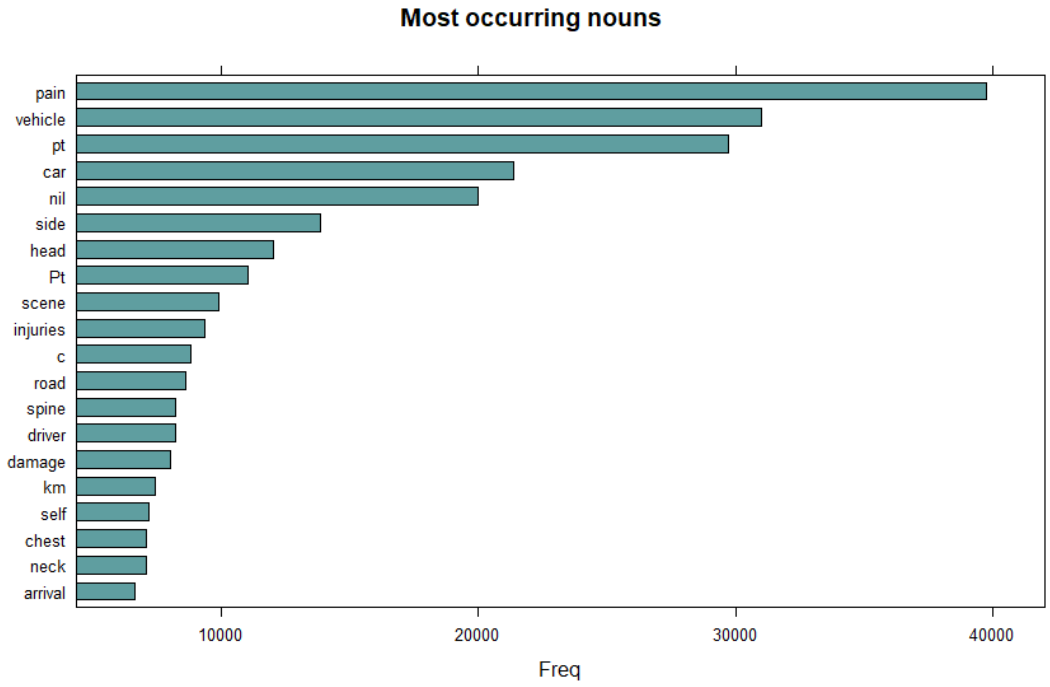

Figure 18. Most Occurring Nouns in QAS Reports (Generated by R)

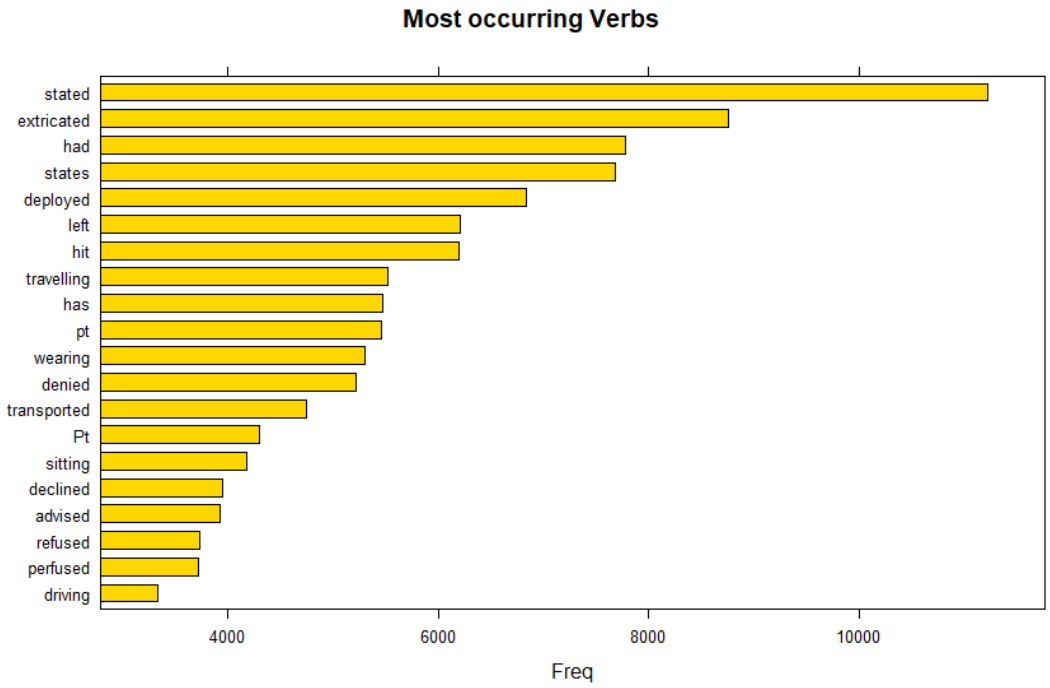

Figure 19. Most Occurring Verbs in QAS Reports (Generated by R)

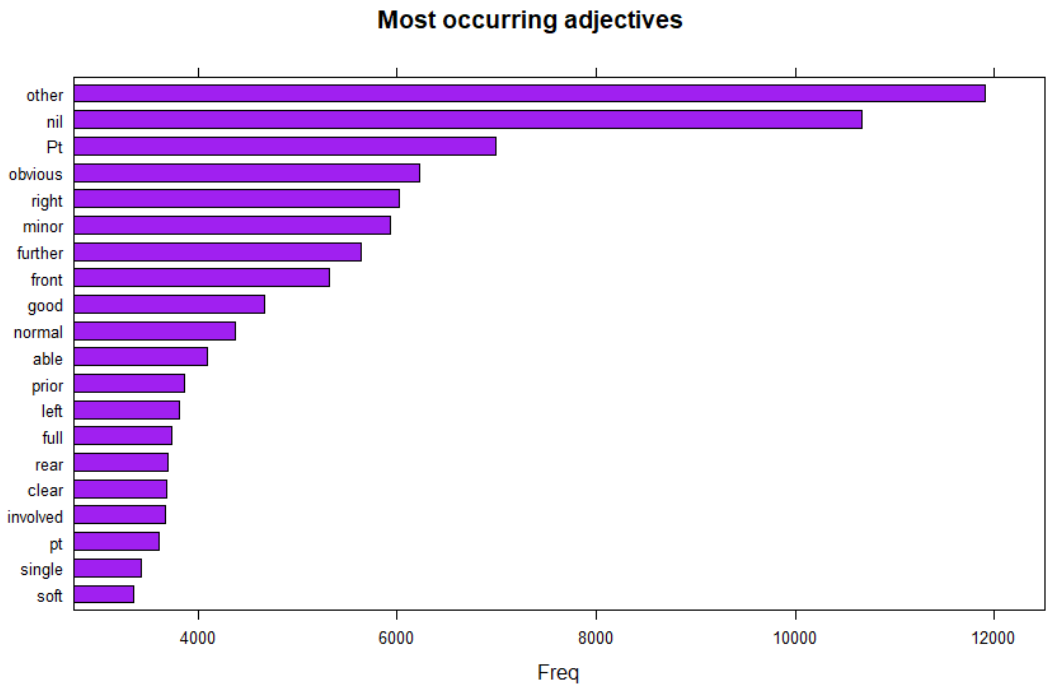

Figure 20. Most Occurring Adjectives in QAS Reports (Generated by R)

4.4.2 Keyword Identification

By identifying the keywords of each sentence, the main idea of each sentence can be briefly represented. Using this keyword representation method can probably reduce the redundancy comparatively. Given the “adjective + noun” pattern, RAKE identified the corresponding keywords based on the whole corpus. The phrases with the top 20 RAKE scores (the possibility that these phrases exist in each sentence) are shown in Figure 21. Some age describing phrases have already been removed from the original figure (including yr old, old male, year old, and years old). Most of the top ranked phrases are commonly used to describe patients’ conditions.

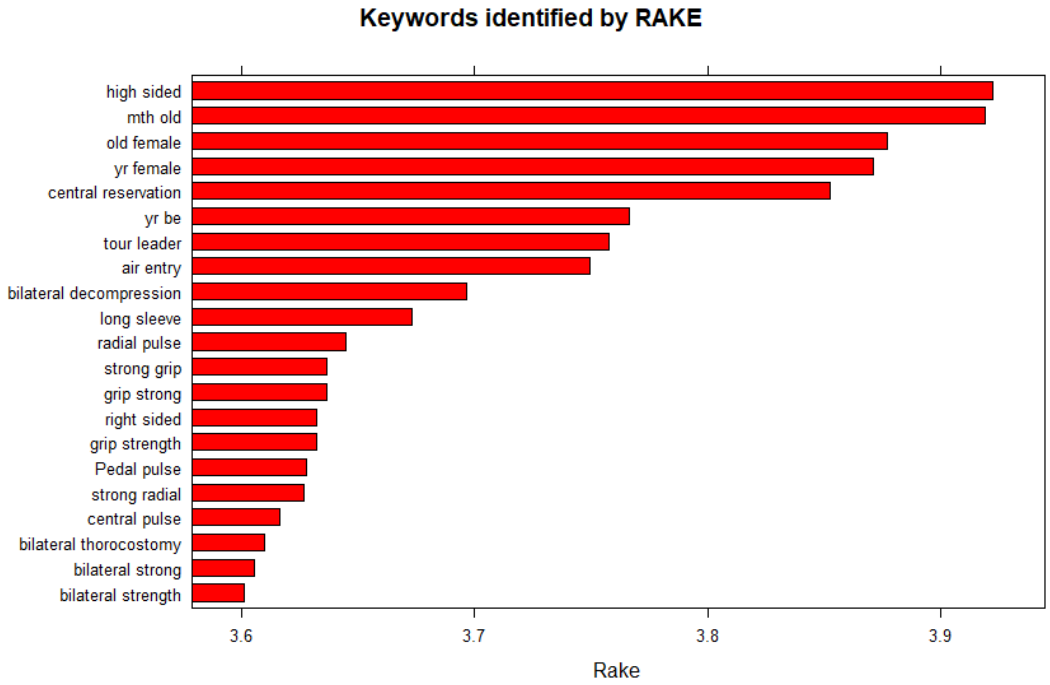

Figure 21. Top 20 Keywords Identified by RAKE in QAS Reports (Generated by R)

Subsequently, RAKE was also used to identify the simple noun phrases ad noun-verb pair phrases. Table 12 shows the identified simple nouns. Because the method works by calculating word co-occurrence to decide how likely a phrase (combined by words) exists. It lists all the possible phrases first as long as the POS meets the required pattern. Thus, for the same content, it might be tokenized several times using different N-Grams. Table 12 shows the final word count for the possible phrases. This

result shows that patient, pain, and nil are the most frequent words in the simple noun phrases. It does not refer to more useful information.

	keyword	ngram	pattern	start	end
1	front seat passenger of vehicle that was	7	ANNPNNV	4	10
2	front seat passenger of vehicle that was hit	8	ANNPNNVV	4	11
3	seat passenger of vehicle that was	6	NNPNNV	5	10
4	seat passenger of vehicle that was hit	7	NNPNNVV	5	11
5	passenger of vehicle that was	5	NPNNV	6	10
6	passenger of vehicle that was hit	6	NPNNVV	6	11
7	vehicle that was	3	NNV	8	10
8	vehicle that was hit	4	NNVV	8	11
9	vehicle that was hit by another car	7	NNVVPDN	8	14
10	that was	2	NV	9	10
11	that was hit	3	NVV	9	11
12	that was hit by another car	6	NVVPDN	9	14
13	was hit by another car	5	VVPDN	10	14
14	hit by another car	4	VPDN	11	14
15	merging lanes	2	VN	16	17
16	glass window broke	3	NNV	23	25
17	glass window broke over pt	5	NNVPN	23	27
18	window broke	2	NV	24	25
19	window broke over pt	4	NVPN	24	27
20	broke over pt	3	VPN	25	27
21	she hit	2	NV	29	30

Figure 12. Simple Noun Phrases Identified by RAKE in QAS Reports (Generated by R)

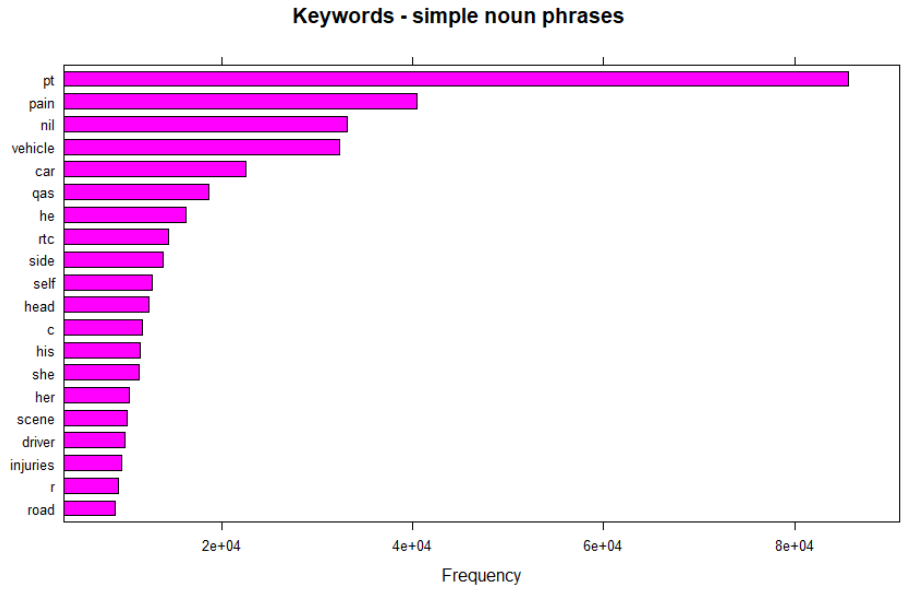

Figure 22. Word Count for the Simple Noun Phrases Identified by RAKE in QAS Reports (Generated by R)

To understand what these noun phrases are in the reports, a bar chart was drawn. As shown in Figure 23, most of the noun phrases identified decribe the patients (patient was; patient stated; patient denied).

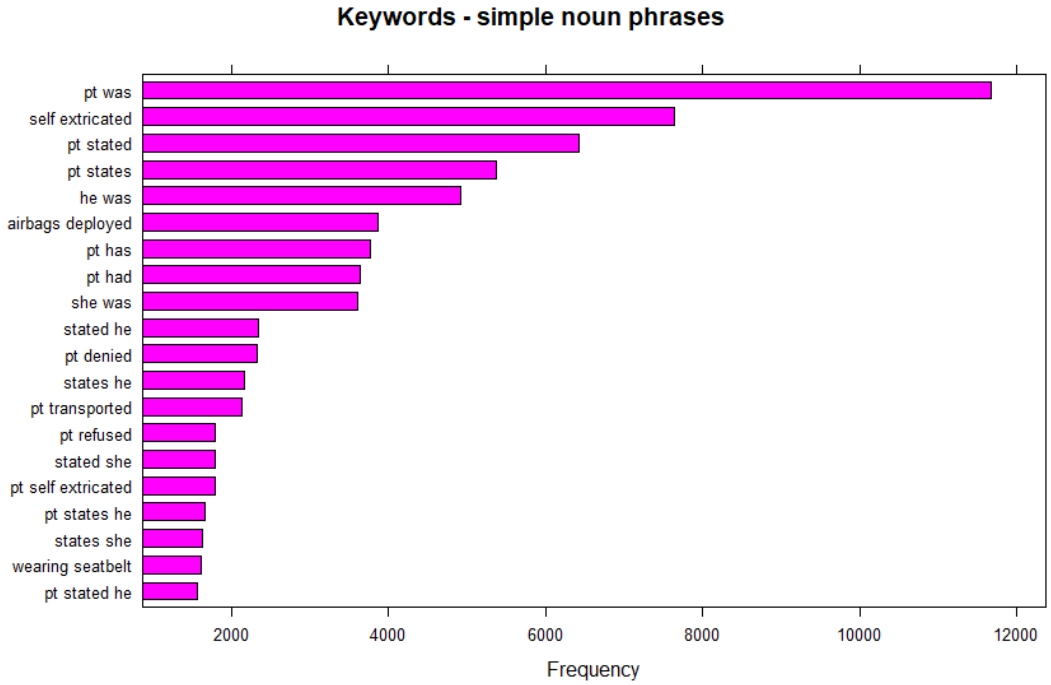

Figure 23. Simple Noun Phrases Identified by RAKE in QAS Reports (Generated by R)

.

4.4.3 Random Forest Model for Keyword TF-IDF

Compared to using the noun-verb phrases, using adjective noun phrases to represent the reports are more appropriate because adjective noun phrases provide more useful information based on the graphs listed above. Thus, after replacing each sentence with its adjective-noun keywords in each document, a keyword TF-IDF was constructed. Based on the keyword TF-IDF, a random forest model was fitted.

The test accuracy of the random forest model is shown as Figure 24. The accuracy is the highest when using no predictors and is consistently low. The accuracy of the model was reduced by adding predictors, which does not make sense for general classifiers. The first possible reason is that a considerable number of redundancies was generated when using different N-Grams to tokenize the same content. Second,

after keyword identification, some common but not useful words contained in the phrases increased for repeating tokenization, but the total number of the report words decreased. Third, there are a number of repeated stop words in this keyword TF-IDF. These make the classifier performs even worse than the previous one. To improve this, we need to develop a full stop word list to tell the machine when to start to tokenize and when to stop. Additionally, we need to set an appropriate N for each different pattern when doing N-Grams tokenization instead of doing it multiple times as the content POS meets the adjective-noun patterns. These suggestions require large amounts of manual time, but by doing these, we can avoid over-tokenization and ensure the identified keywords can largely represent the original content.

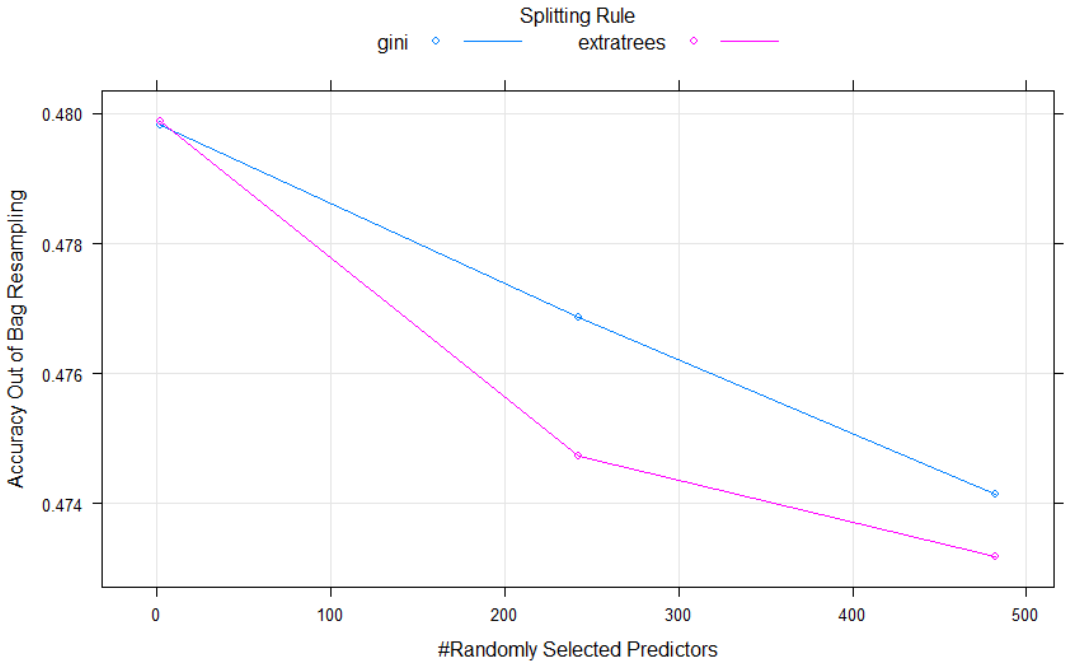

Figure 24. The Random Forest Test Accuracy Using Different Splitting Rules and Parameters Based on Identified Simple Noun Phrases (Generated by R)

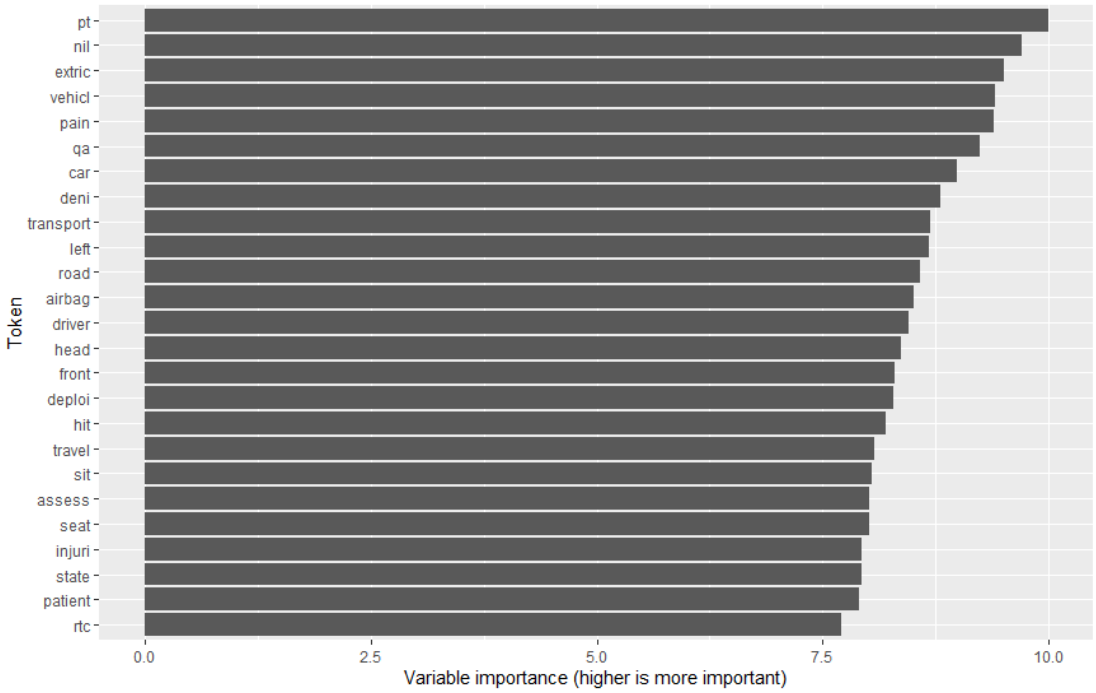

Figure 25. The Variable Importance Ranking of Random Forest Based on Identified Simple Noun Phrases (Generated by R)

4.5 Word Vector Representation

The whole data set was used to train the word vector (word embedding) model. The reasons for not using an existing trained word vector model are:

- There are lots of specific terms and abbreviations in the QAS data, using an external model might result in large distances among similar or same meaning terms.
- The QAS data is a large enough corpus to train a word vector model. QAS reports are written in the same style. Using the QAS corpus to build a model can easily determine which words describe similar content.

In the word vector, similar words are clustered together (having low distances to each other). After dimensionality reduction by T-SNE, the word vector two-dimension space was visualized as Figure 26 and Figure 27. They are the same vector showing the most frequent 2500 and 1000 words respectively. Figure 26 looks crowed; some words are covered by each other. Figure 27 is more user-friendly; the words are clearly displayed comparatively.

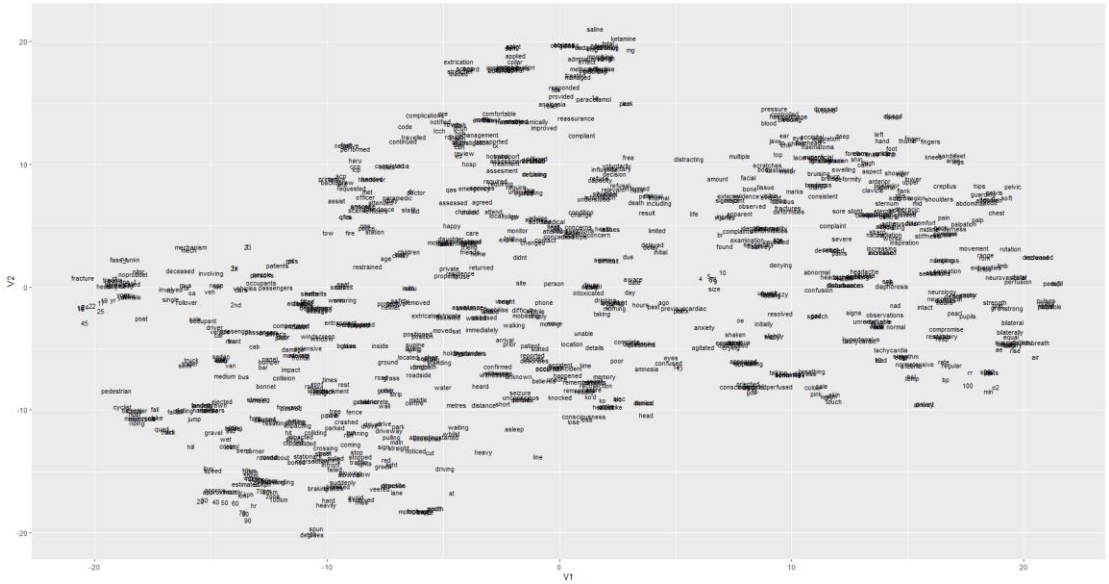

Figure 26. The Word Vector Dimensionality Reduction Result (Only showing the most frequent 2500 words. Generated by R)

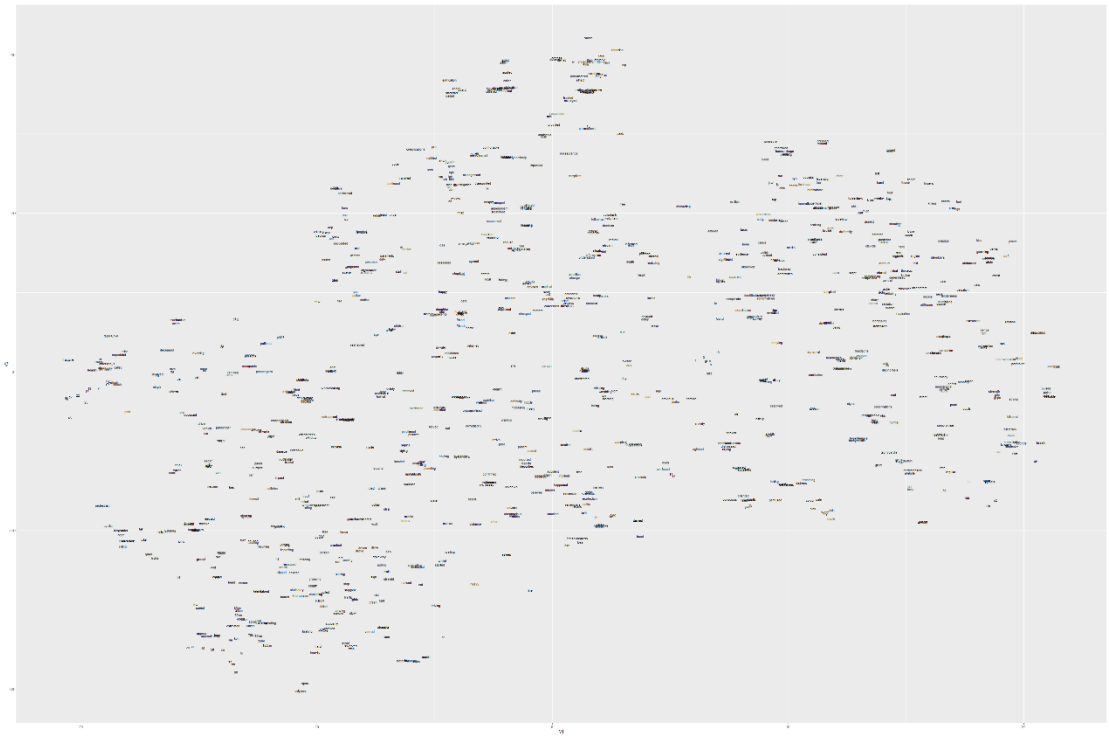

Figure 27. The Word Vector Dimensionality Reduction Result (Only showing the most frequent 1000 words. Generated by R)

By zooming into the details of Figure 27, the partial word clusters can be visualized. As the human face organs and normal injuries on human faces were closed to each other in the generated word vector, the word vector model is proven working well. The word vector can also address the abbreviation problem as mentioned in the challenges section. As Figure 28 grouped the hospital abbreviations as a cluster and the “years old” abbreviations as another cluster, this word vector can help us to identify the abbreviations which refer to the similar or same content.

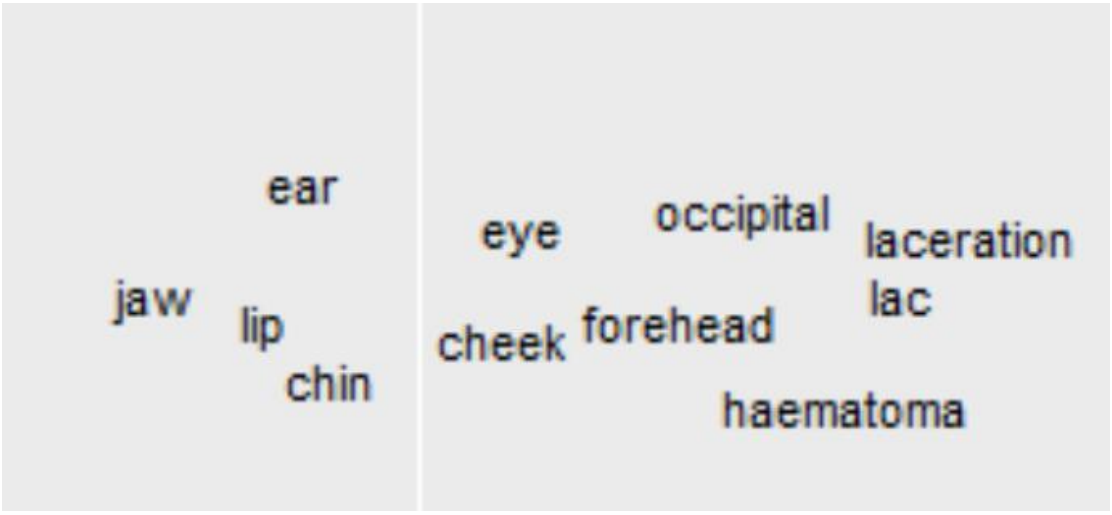

Figure 28. The Zoom in Details 1 of Word Vector Dimensionality Reduction 1000 Words Result (The Human Facial Organs and Normal Injuries on Human Faces)

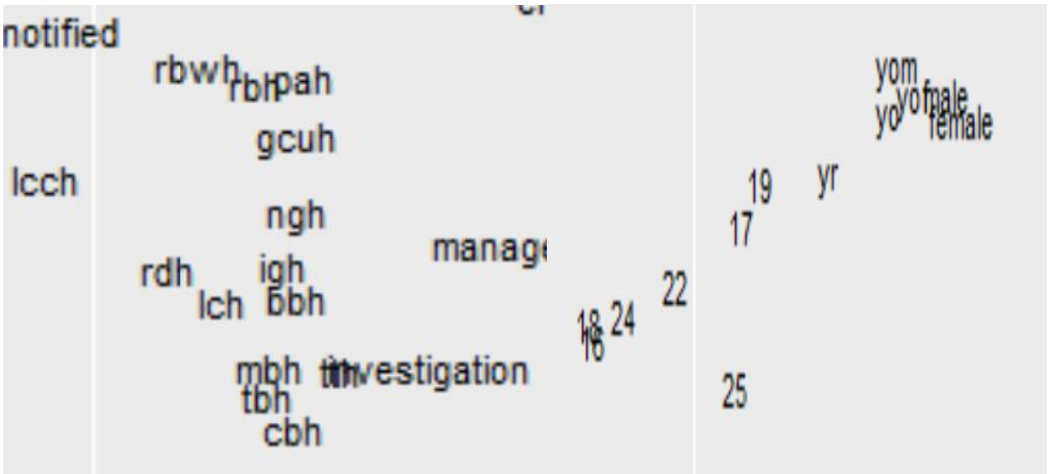

Figure 29. The Zoom in Details 2 of Word Vector Dimensionality Reduction 1000 Words Result (The Abbreviations of The Hospitals and The Most Frequent Patient Age)

4.5.1 Document Vector Representation

By adding each word vector for each document in the whole dataset, a document vector matrix can be generated. This document vector matric can roughly represent the overall meaning of each report's content in a high dimensional space. Thus, using this vector to classify the reports can avoid the issue that the reports were not organized in natural English and using several abbreviations to represent the same word or phrase (as mentioned in the Challenges section).

After the document vector is constructed, T-SNE was applied to reduce the dimensional space for visualization and reducing the model complexity of the SVM classifier. The reasons why the document vector and T-SNE were constructed and calculated using the whole dataset are:

- If a word vector is trained using a small sample, the result might not be accurate (similar words might not be clustered together).

- T-SNE generates a unique high dimensional space for each different input. The reduced two-dimensional space can represent different meanings because of the different input. Thus, an SVM model trained by T-SNE result of the training data is not applicable for T-SNE result of the test data or the validation data

The reports were separated to two large clusters after dimensionality reduction as Figure 30 shown. The larger cluster represents the reports of the patients without serious problems. These patients are mostly well-being after the car accidents and do not need to transfer to the hospitals. While the smaller cluster refer to reports of serious car accidents with patients get injured. The reason why there is a such obvious boundary between these two clusters is the document vector constructing method used. Because each word vector was summed up for each document, the results are significantly different between reports with different length. The reports

of slight crashes are normally short while the reports of serious crashes are comparatively longer. To remove the length influences, the document vector was also built using the average word vectors in each report as shown in Figure 31. In both Figure 30 and Figure 31, reports of different causes are evenly mixed, this indicates that the SVM model based on this might not work efficiently.

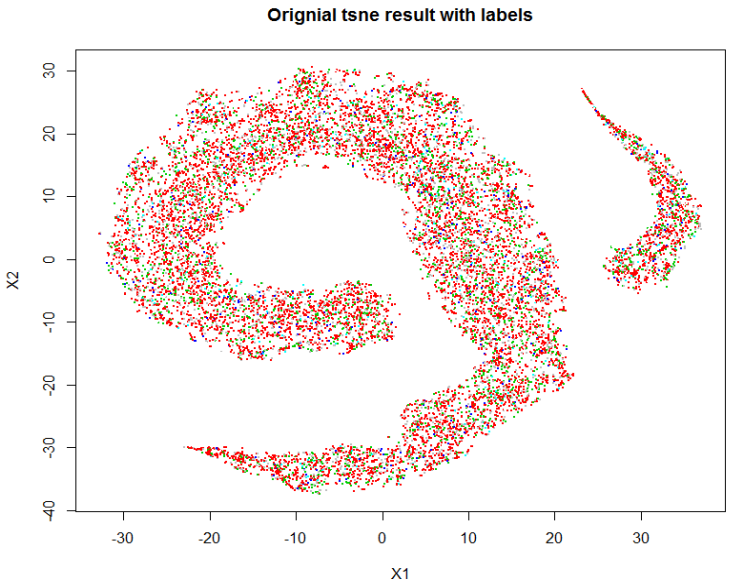

Figure 30. The Sum Document Vector Dimensionality Reduction Result Using the Whole Dataset. Green color stands for stationary object collisions (2667 cases); Red color stands for motor vehicle collisions (8151 cases); Pink color stands for non-collisions (2420 cases); Light Green stands for collisions with pedestrian or animals (324 cases); Blue stands for other unspecified collisions (398 cases); Black stands for other collisions (1337)) (Generated by R)

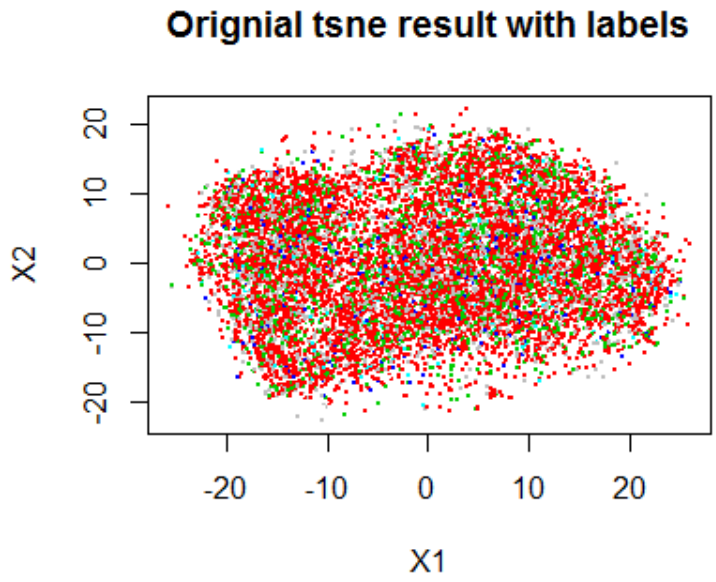

Figure 31. The Mean Document Vector Dimensionality Reduction Result Using the Whole Dataset. Green color stands for stationary object collisions (2667 cases); Red color stands for motor vehicle collisions (8151 cases); Pink color stands for non-collisions (2420 cases); Light Green stands for collisions with pedestrian or animals (324 cases); Blue stands for other unspecified collisions (398 cases); Black stands for other collisions (1337)) (Generated by R)

4 .5.2 SVM for Document Vector

Based on the two T-SNE results, two SVM models were built. Figure 32 and Figure 33 indicates the SVM classification results on the whole dataset. For the first SVM model (based on the sum document vector), the test accuracy was 59%, while for the second one (based on the mean document vector), the test accuracy was 62%, slightly higher than the first one. Thus, the latter one was evaluated using a confusion matrix. As the confusion matrix is shown (Figure 34), the SVM model performed badly when classifying reports caused by collisions with others, other unspecified collisions, stationary object collisions, non-collision, and collisions with pedestrian or animals. The accuracy of the classifier looks satisfied only because of the large number of cases caused by motor vehicle collisions.

Figure 32. The Sum Document Vector SVM Classified Result (Generated by R) Green color stands for stationary object collisions (3635 cases); Red color stands for motor vehicle collisions (9701 cases); Pink color stands for non-collisions (2587 cases); Light Green stands for collisions with pedestrian or animals (120 cases); Blue stands for other unspecified collisions (10cases); Black stands for other collisions (118))

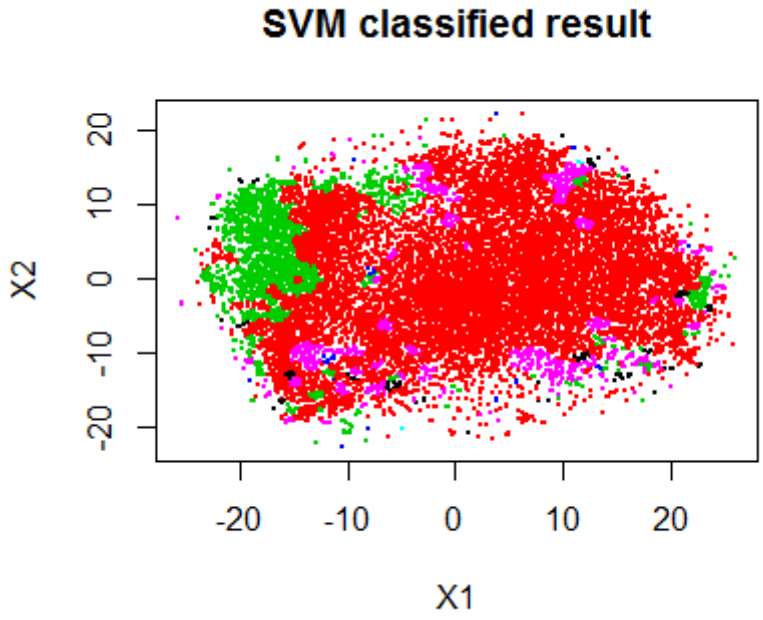

Figure 33. The Mean Document Vector SVM Classified Result (Generated by R) Green color stands for stationary object collisions (3635 cases); Red color stands for motor vehicle collisions (9701 cases); Pink color stands for non-collisions (2587 cases); Light Green stands for collisions with pedestrian or animals (120 cases); Blue stands for other unspecified collisions (10cases); Black stands for other collisions (118))

Confusion Matrix and Statistics

Prediction	Reference: Coll-Other	Mot-Coll	Non-Coll	Other	Ped-A	Stat-Obj
Coll-other	17	177	33	0	0	15
Mot-Coll	6	1567	61	1	0	46
Non-Coll	2	226	199	1	0	29
Other	1	44	12	7	0	3
Ped-A	2	47	19	0	2	8
Stat-Obj	5	366	55	0	0	99

Statistics by Class:

	Coll-Other	Mot-Coll	Non-Coll	Other	Ped-A	Stat-Obj
Sensitivity	0.515152	0.6457	0.52507	0.777778	1.0000000	0.49500
Specificity	0.925423	0.8170	0.90341	0.980270	0.9750656	0.85053
Pos Pred Value	0.070248	0.9322	0.43545	0.104478	0.0256410	0.18857
Neg Pred Value	0.994302	0.3718	0.93058	0.999330	1.0000000	0.96000
Prevalence	0.010820	0.7957	0.12426	0.002951	0.0006557	0.06557
Detection Rate	0.005574	0.5138	0.06525	0.002295	0.0006557	0.03246
Detection Prevalence	0.079344	0.5511	0.14984	0.021967	0.0255738	0.17213
Balanced Accuracy	0.720287	0.7313	0.71424	0.879024	0.9875328	0.67276

Figure 34. The Confusion Matrix and Model Evaluation Indicators for the SVM Model Based on the Mean Document Vector

5.Disscussion

This project used several methods to classify the reports in order to explore which methods are suitable for the QAS data and determine how to ultimately convert the QAS data into structured data. In this part, the performances of the model are discussed and the suggestions for further analysis are provided.

5.1 Models Performances

5.5.1 Topic model

The project tried to classify the reports into different topic using the LDA topic model. Because how many topics in all the reports is not known, a range of parameters (Topics from 4 to 20, and Terms from 7 to 12) were used to build a few LDA models. However, the results turn out not being good, all the models' topics can only explain 3% - 6% of the reports.

Because the QAS reports are very detailed but in a similar writing pattern. It is

difficult for LDA with a limited number of topics to assign topics and documents together. Even we using a considerable number of topics to moderate the model, the topic model can still not perform in a satisfactory level as most of the cases are telling a same general topic (which is accidents and patients) but different in details which can be unique (not enough training samples). Only a few words in the reports can reflect the uniqueness of the reports, most of the other words are about the same general topics. Therefore, using the topic model to classify the QAS reports is verified not being appropriate.

5.5.2 TF-IDF and Random Forest

The TF-IDF based Random Forest has the highest accuracy among all the models used in the project because the TF-IDF can reflect the word composition and word importance in a report. However, it can still be improved because the TF-IDF not only provides the causes information.

5.5.3 Keyword Identification

This project identified the frequent adjective-noun pairs in the reports as the keywords and used these keywords to represent sentences. This implement can reduce the redundancy of the original reports. This representation method only provides the keywords for each sentence. However, the original reports were not organized in natural English, this method might not necessarily be accurate. Moreover, this keyword identification method lists all the possible tokens as long as the tokens meet the required patterns even though these tokens are from the same content. This results in more redundancy when using these keywords to represent the reports.

5.5.4 Document Vector and SVM

The document vector-based SVM performed unsatisfactorily because of the same reason of TF-IDF based random forest's bad performance. Both TF-IDF and the document vector represent the overall content of each report, which provides too much information. Using irrelevant information does not help to improve the performance of a classifier. However, the SVM model did classify the slight crash cases from the serious crash cases.

5.2 Suggestions

For the previous method used, the classifiers' unsatisfactory performances both because of the inappropriate representation methods. These representations either giving too much information or too little. Thus, we might need to divide the reports into segments. For instance, to classify the causes of the accidents, the sentences that describe causes should be firstly extracted from each report, subsequently, we can build a classifier based on these sentences.

To identify these sentences, the RAKE and word vector we mentioned can be used. Firstly, a list of reasonable N-Grams tokenization rules shall be generated (including where to start and where to end, how to deal with stop words, and how to avoid the same tokens being tokenized several times). Based on these rules, a representing keywords list might be generated.

Secondly, using this word vector, the keywords can be clustered to a certain number of groups. Subsequently, we will be able to define the category of each keyword group so that we can identify what content each sentence is about based on the keywords. For instance, if we identified a number of keywords (including "pedal pulse", "bilateral decompression", "radial pulse", "window broke", and "merging

lanes”.) After that, these words can be clustered into 2 groups based on the word vector. “Pedal pulse“, ” bilateral decompression“, and ”radial pulse“ belong to the patients’ conditions and ”window broke“ and ”merging lanes“ belong to the causes of the accidents. Thus, for each of the sentences whose keywords belong to any defined cluster, it is classified as a sentence describing the relevant content (A sentence whose keywords are “merging lane” is a sentence about causes).

Thirdly, after we classify the sentences, we can classify the reports only based on the selected sentences instead of the overall content. For example, if we need to classify the causes of the accidents, build a classifier (no matter SVM or random forest) only using the sentences about causes. This will significantly improve the model performance

6.Conclusion

The project identified the areas and road sections of frequent traffic accidents in Queensland based on the QAS data. For these areas and sections, the Department of Transportation can appropriately improve the road setting. In addition, the project used classifications as an example to explore how to effectively use text analysis methods to convert QAS unstructured data into structured data. However, the models built are not satisfactorily performed. For the peculiarity of the QAS data, the point of this conversion is how to reasonably express the text content into a digital form. To achieve this, the project also provided suggestions for further study on the QAS data.

7.Reference List

4991094 Method for language-independent text tokenization using a character categorization. (1991). *Expert Systems with Applications*, 3(3), p.VI.

A New Computational Model of Language Development and Language Processing. (2012). *Sensorimotor Cognition and Natural Language Syntax*. doi:10.7551/mitpress/8938.003.0008

Anderson, M. and Vilares, D. (2018). Increasing NLP Parsing Efficiency with Chunking. *Proceedings*, 2(18), p.1160.

Arumugam, M. (2019). Processing the Textual Information Using Open Natural Language Processing (NLP). *SSRN Electronic Journal*.

Barber, A. (2000). A pragmatic treatment of simple sentences. *Analysis*, 60(4), pp.300-308.

B. Chrystal, J., & Joseph, S. (2015). Text Mining and Classification of Product Reviews Using Structured Support Vector Machine. *Computer Science & Information Technology (CS & IT)*. doi:10.5121/csit.2015.50803

Bradley, C. (1984). 68.32 Kendalls Correlation Coefficient Revisited. *The Mathematical Gazette,68*(445), 214. doi:10.2307/3616352

Boroš, M. and Maršík, F. (2012). Multi-Label Text Classification via Ensemble Techniques. *International Journal of Computer and Communication Engineering*, pp.62-65.

Chen, Z., Huang, Y., Liang, Y., Wang, Y., Fu, X. and Fu, K. (2017). RGloVe: An Improved Approach of Global Vectors for Distributional Entity Relation Representation. *Algorithms*, 10(2), p.42.

Cilliers, F. (1992). Distributed Representation: A critique of minimal networks. *Artificial Neural Networks*,1391-1394. doi:10.1016/b978-0-444-89488-5.50121-4

Classification: Practice - Random Forest. (2018). doi:10.4135/9781526469144

Common Text Mining Visualizations. (2017). *Text Mining in Practice with R*, 51-83. doi:10.1002/9781119282105.ch3

Evangelopoulos, N. (2013). Latent semantic analysis. *Wiley Interdisciplinary Reviews: Cognitive Science*, 4(6), pp.683-692.

Harris, Z. (1954). Distributional Structure. *WORD*, 10(2-3), pp.146-162.

Hassler, M., & Fliedl, G. (2006). Text preparation through extended tokenization. *Data Mining VII: Data, Text and Web Mining and Their Business Applications*. doi:10.2495/data060021

Hearst, M. A. (2012). Text Data Mining. *Oxford Handbooks Online*. doi:10.1093/oxfordhb/9780199276349.013.0034

Ishihara, S. (2014). A Comparative Study of Likelihood Ratio Based Forensic Text Comparison Procedures: Multivariate Kernel Density with Lexical Features vs. Word N-grams vs. Character N-grams. *2014 Fifth Cybercrime and Trustworthy Computing Conference*. doi:10.1109/ctc.2014.9

Kivenko, I. (2018). THE INTEGRATION OF GRATITUDE COMMUNICATIVE MOVES INTO ENGLISH LITERARY DIALOGUE DISCOURSE. *Odessa linguistic journal,* 11, pp.37-43.

Leskovec, J., Rajaraman, A., & Ullman, J. D. (n.d.). Frequent Itemsets. *Mining of Massive Datasets,*191-227. doi:10.1017/cbo9781139924801.007

Motor vehicle traffic crashes as a leading cause of death in the U.S., 1994. (1998). *PsycEXTRA Dataset*. doi:10.1037/e446782008-001

Pereira, F. and Grosz, B. (1994). *Natural language processing*. Cambridge, Mass.: MIT Press, p.3.

Queensland Ambulance Service. (n.d.). Retrieved from https://www.ambulance.qld.gov.au/index.html

Queensland Road Crash Weekly Report - WebCrash 2.3. (2019, April). Retrieved from https://www.webcrash.transport.qld.gov.au/webcrash2/external/daupage/weekly/roadsense.pdf

Ramm, A. G. (2015). Representation of vector fields. *Global Journal of Mathematical Analysis,3*(2), 73. doi:10.14419/gjma.v3i2.4577

Rizzo, M. L. (2007). Statistical Computing with R. doi:10.1201/9781420010718

Road traffic injuries. (2018). Retrieved from https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

Schonlau, M. and Guenther, N. (2016). Text Mining Using N-Grams. *SSRN Electronic Journal*.

Silge, J., & Robinson, D. (2019, March 23). Text Mining with R. Retrieved from https://www.tidytextmining.com/

Studies in linguistic analysis. (1957). Oxford: Firth.

Text Summarization. (2016). *Text Data Management and Analysis: A Practical Introduction to Information Retrieval and Text Mining*,317. doi:10.1145/2915031.2915048

Topicmodels: An R Package for Fitting Topic Models. (n.d.). Retrieved from https://www.jstatsoft.org/article/view/v040i13/v40i13.pdf

Vens, C., & Costa, F. (2011). Random Forest Based Feature Induction. *2011 IEEE 11th International Conference on Data Mining*. doi:10.1109/icdm.2011.121

Wang, H., Li, Z., & Cheng, Y. (2008). Weighted Latent Dirichlet Allocation for Cluster Ensemble. *2008 Second International Conference on Genetic and Evolutionary Computing*. doi:10.1109/wgec.2008.60

Waskom, M. L., & Wagner, A. D. (2016). Distributed representation of context by intrinsic subnetworks in prefrontal cortex. doi:10.1101/074880

Wensen, L., Zewen, C., Jun, W., & Xiaoyi, W. (2016). Short text classification based on Wikipedia and Word2vec. *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*. doi:10.1109/compcomm.2016.7924894

Wickham, H. (2009). Ggplot2. doi:10.1007/978-0-387-98141-3

Zhang, X. and Wang, T. (2010). Topic Tracking with Dynamic Topic Model and Topic-based Weighting Method. *Journal of Software*, 5(5).

Zhang, Z. (2010). Basic Data Processing Methods in LDA Measurements. *LDA Application Methods*,47-52. doi:10.1007/978-3-642-13514-9_5

Zhao, H., Bai, C. and Zhu, S. (2010). Automatic Keyword Extraction Algorithm and Implementation. *Applied Mechanics and Materials*, 44-47, pp.4041-4049.

ZHU, G. and SUN, W. (2013). Rapid speech keyword spotting method based on template matching. *Journal of Computer Applications*, 33(11), pp.3138-3140.