Hospital data analytics for business intelligence - An analytics tool for patient feedback analysis.

Jeffrey Alan Ray
Department of Computer Science

in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of
Computer Science

Edge Hill University
Ormskirk, England
April 2020

Hospital data analytics for business intelligence - An analytics tool for patient feedback analysis.

Abstract

Hospitals gather huge amounts of valuable data every day, from clinical studies to patient satisfaction surveys. However, the full utilisation of this data is somewhat lacking with limited data analysis taking place. Improving hospitals data analysis can facilitate the decision-making process leading to informed decisions that will positively influence the patient's journey. In collaboration with Alder Hey hospital this research identified that the processing time of patient satisfaction data often took 6 to 8 months for analysis. Patients often failed to see any change after making a suggestion or comment, this thesis addresses this issue by utilising machine learning and data visualisation techniques to help reduce the processing time for analysis of text-based patient feedback. This research focuses on the integration of techniques based on machine learning, neural networks, mathematical modelling(Topological data analysis) and data visualisation to actively assess real-time feedback from Twitter and stored textual data (the Family and Friends Test). To allow greater analysis in a much-reduced time frame. The thesis demonstrates such integration via the creation of a data analytics tool programmed in Python. Existing approaches require significant computational power to extract the textual information this thesis demonstrates timely extraction with low powered computing systems. Existing solutions do not widely integrate all the techniques used in this thesis, making this research unique by combining the approaches in a low

powered computing environment. The research demonstrates the effectiveness of small-scale Machine learning running on low powered computing systems when applied to a text-based extraction and analysis. The research contributes novel algorithms using Topological data analysis and Network theory. The application of this research will aid researchers and business in the utilisation of machine learning and data extraction to support business intelligence gathered from text sources.

Contents

Listing of figures

Published Papers

Works Published from this thesis

J. Ray and M. Trovati, "A Survey of Topological Data Analysis (TDA) Methods Implemented in Python," in INCoS, Springer International Publishing, 2017, pp. 594–600.

J. Ray, O. Johnny, M. Trovati, S. Sotiriadis, and N. Bessis, "The Rise of Big Data Science: A Survey of Techniques, Methods and Approaches in the Field of Natural Language Processing and Network Theory," Big Data and Cognitive Computing, vol. 2, no. 3, p. 22, Aug. 2018.

J. Ray and M. Trovati, "On the Need for a Novel Intelligent Big Data Platform: A Proposed Solution," in Advances in Intelligent Networking and Collaborative Systems, Springer International Publishing, 2018, pp. 473–478.

J. Ray, M. Trovati, and S. Minford, "A Preliminary Automated Approach to Assess Hospital Patient Feedback," in Advances in Intelligent Networking and Collaborative Systems, Springer International Publishing, 2017, pp. 585–593.

"An Entropy Based Approach to Real-Time Information Extraction for Industry 4.0," M. Trovati, H. Zhang, J. Ray and X. Xu, in IEEE Transactions on Industrial Informatics. 2019 (https://doi.org/10.1109/TII.2019.2962029)

”DLCD-CCE: A Local Community Detection Algorithm for Complex IoT Networks,” X. Xu, N. Hu, M. Trovati, J. Ray, F. Palmieri and H. M. Pandey, in IEEE Internet of Things Journal. 2019 (https://doi.org/10.1109/JIOT.2019.2960743)

“GORTS: genetic algorithm based on one-by-one revision of two sides for dynamic travelling salesman problems,” X. Xu, H. Yuan, P. Matthew, J. Ray, O. Bagdasar, and M. Trovati, Soft Computing, Sep. 2019 (https://doi.org/10.1007/s00500-019-04335-2)

Acknowledgments

I would like to thank my director of studies Dr. Marcello Trovati, without his valuable guidance, knowledge and support this would not be possible.

I would also like to thank my supervisory team, Dr. Huaizhong Zhang and Dr. Peter Matthew my deepest gratitude to you both.

And to the staff at Alder Hey Children's Hospital Innovation Centre and those at Angels Data.

1 Introduction

Data is continuously created every day, capturing human activities, health information and sensor data, to name but a few. In particular, the health sector has become increasingly reliant on such data, which has enormous potential in the identification and assessment of actionable information from clinical studies to patient satisfaction surveys. This would provide healthcare professionals with important tools to facilitate real-time knowledge discovery i.e. diagnosis, or research-based. Furthermore, enhancing hospitals data analysis is likely to make a substantial contribution to the decision-making process creating informed decisions that will positively influence the patient's journey.

The purpose of this thesis is to generate an efficient tool, which combines machine learning, natural language processing, network theory and mathematical modelling to process a large amount of available social media data relating to the hospital. The main objective is to turn such wealth of unstructured data into actionable infor-

mation, which improves the overall patients' experience and healthcare provision whilst within the care of the hospital. This has the potential to reduce patient waiting time or provide specialist staff who relate to identified trends from social media. The project and research involve working with Alder Hey Children's Hospital Innovation Hub, a dedicated team who aim to solve real-world healthcare challenges with cutting edge technology and Angels Data, a data and lead generation company who specializes in lead sourcing for outbound telemarketing as two organizations who require automation and machine learning to improve processing times and produce meaningful analytics and business leads. This work investigates current machine learning theories and techniques that would help analyse social media free-form text entries to improve the quality of healthcare provided at the hospital. Also, the system enables any social media feed to be analyses and provide additional insight into trends and patterns of social data. This research demonstrates how a combination of novel techniques merged with existing technologies creates opportunities for real-time data processing and analytics to significantly speed up and improve the quality of data outcomes when applied to the available datasets, for current methods applicable to this context.

This chapter gives detail on the research objectives, the Data collection and data storage of research material, the original contribution to knowledge made by this thesis, the motivation for the research and provides a graphic overview of the thesis structure.

1.1 Research objectives

1. To find the data analytics most suitable for providing usable information to hospitals and identify what criteria is used to assess these tools. With a large amount of available data, it is important to ensure that the information extracted by the process provides meaningful outcomes, with a direct impact on patients' experience along with the accessibility of the data. Poorly analysed data could cause a hindrance to the knowledge management within the hospital with a negative and measurable influence on the overall user's expe-

rience. The wide variety of machine learning models that could be utilised to interpret the data, have been carefully considered and discussed initially in the Literature Review section 2.1 with the specific approach further detailed in chapters 4 and 5

2. To investigate ways to enhance these data analytics tools and technologies. The suitable methods that are used to create the application allow the ability to fine-tune the system to ensures that a quality outcome is produced. Creating a bespoke analytic tool allows for highly detailed and informative knowledge directly relevant to the hospital and patient. The suitability and feasibility of the system take into account hardware considerations and limitations that influence a fully optimised solution. The process of fine-tuning involved branching the system away from hospital data towards business analytics and social discovery as part of a smaller research project for an international business Angels Data.

3. To enhance the ability of data analysis, processing and information resulting to provide the hospital users with the best functional information and experience. The aim for an overall improvement of the patients' experience, is a large and daunting task however, it is possible to deliver timely feedback and provide patients with the ability to feel their views and opinions are being heard and acted upon. Chapter 4 demonstrates initial work to process opinions and create actionable information from the family and friends test. This analysis often takes 6 to 8 months of processing time, the automation of the process allows change to happen so that patients receive information that is both valuable, in an accessible format, and importantly delivered promptly. Providing the end-users with a wealth of unstructured data adds no value to the experience, and may result in reduced co-operation between patients and staff. With the demonstration of enhanced data analysis and visualisation, administrator and management can create change, which enables informed decisions and increases patient awareness of medical or ancillary information that may be relevant to treatment.

4. To assess how advancing healthcare platforms can help accelerate data analytics for hospitals. With the continuous advance of healthcare platforms, more data will be produced within the hospital. Therefore, the ability to integrate this information into modern analytics set is at the core of this research, the use of the open-source solution and to create new algorithms and methods to create an open platform for utilising machine learning. Within machine learning and AI, the progression of the information and the associated technology is remarkable, the system created and results in chapter 6. It is clear from chapter 4 that the data can be immediately utilised and made accessible across the hospital, allowing important decisions to be made and backed up by supporting data. A larger data pool allows increased analytics to be performed that may result in new patterns and discoveries that benefit both patient data access but also medical professionals.

1.2 Data Collection

The data from a single months Family and Friends Test and tweets have been aggregated. Following data protection regulations, no privacy issues have arisen. All data from Alder Hey Children's hospital have been anonymised with full cooperation and scrutiny of the hospital patient data security team to ensure confidentiality. The collected data has been stored on the secure Edge Hill university network, in a password-protected .zip file. Despite none of the data being of a sensitive or critical nature (nor does it contain any personal information), the best practice guidelines of the university are diligently followed to maintain a secure and risk-free data storage environment, see [124] for data management policy.

1.3 Original contribution to knowledge

The purpose of this work is to generate an efficient analytics tool, which combines machine learning, Natural Language Processing, Network Theory and mathematical modelling to process a large amount of available hospital data. The main ob-

jective is to transform such large unstructured data into actionable information, which will result in the potential improvement of the overall patients' experience and healthcare provision, whilst within the care of the hospital.

The thesis demonstrates how better integration of data extraction from various sources, such as blogs, social media and general web resources can be used to supplement the decision-making process.

The thesis contributes towards the application of machine learning in a business environment, via the integration of machine learning and topological data analysis. Transforming theoretical machine learning methods and with the creation of novel algorithms and methods creates an application of machine learning and text analysis in python.

The methods utilised here have not been widely integrated before this research. The thesis has demonstrated that given modern open-source libraries it is entirely possible to create machine learning systems that can utilise a low powered system (no GPU assistance) to create real-time machine learning assisted analytics from social media as well as processing collected data from hospital sources.

1.4 Motivation

This research has been motivated by discussions with Alder Hey Children's Hospital regarding the use of patient feedback and its slow response to implementing change. On average 6 to 8 months elapsed between feedback being received and action taken. A reduction in this time via the use of machine learning aims to positively influence the patients' journey throughout the hospital setting.

1.5 Thesis Overview

This thesis is divided into 7 chapters, the introduction sets out the objectives of the thesis along with a general overview of the research.

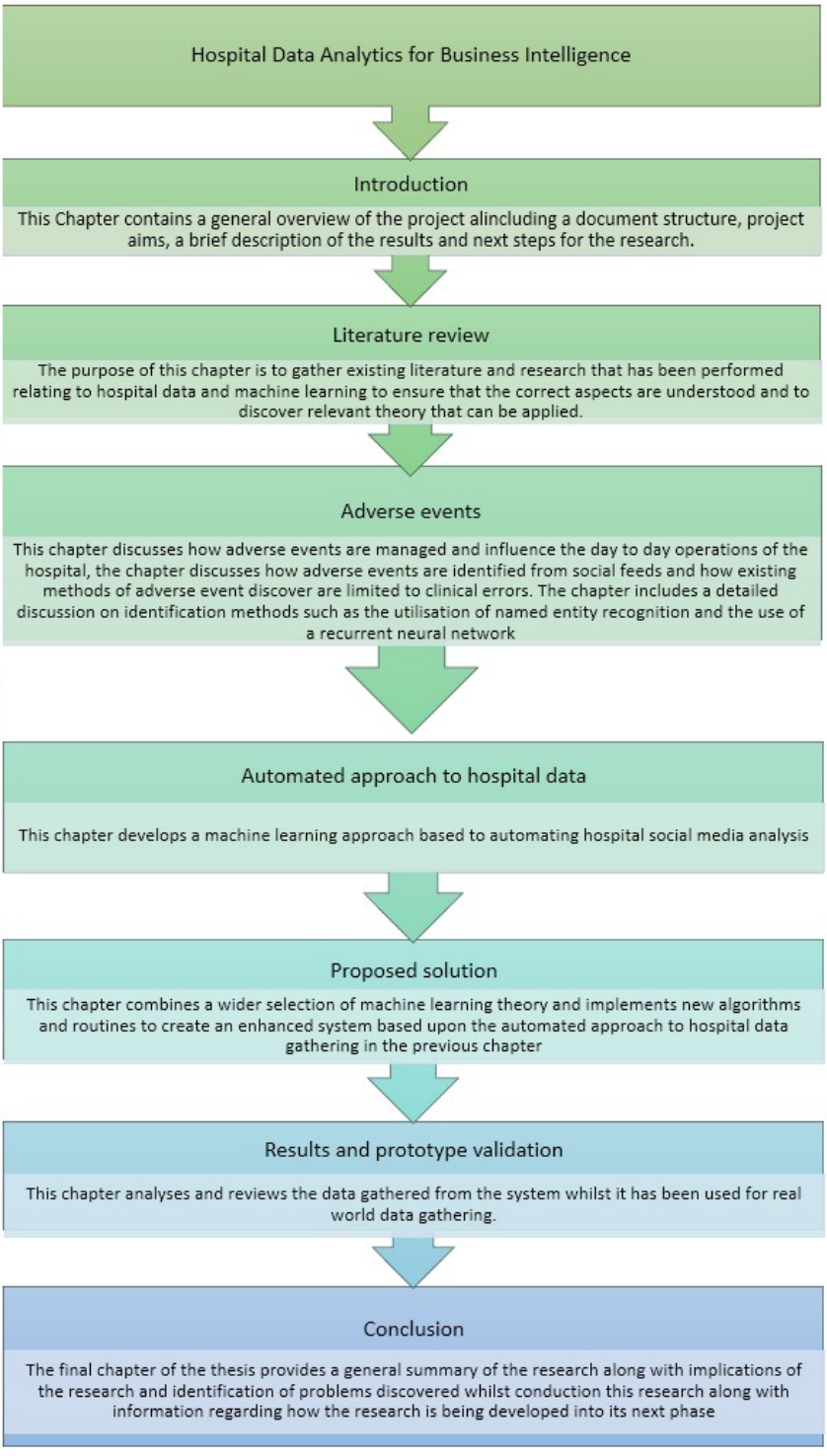

Figure 1.5.1: Thesis overview

2

Background & Literature Review

The purpose of this chapter is to gather and understand existing literature and research that has been performed relating to patient feedback and machine learning to ensure that the correct aspects are understood and to discover what relevant theory that can be applied. The chapter consists of A background on hospital big data and the important factors for using patient feedback successfully, A discussion on the main existing approaches created by the major technology companies and how these approaches are different from the one utilised in this thesis.

2.1 Background

Big Data is defined as "large volumes of high velocity, complex, and variable data that require advanced techniques and technologies to enable the capture, stor-

age, distribution, management and analysis of the information" [31]. Big data includes features such as multiplicity, rate and veracity when it comes to dealing with healthcare institutes [26]. Hospital data size is likely to increase intensely in the upcoming years [31]. Some hospital compensation models have been fluctuating due to the effect of new factors affecting them such as significant use and "pay for" [94]. Healthcare establishments need to use the existing devices, structure and methods to control big data efficiently or else a large amount of money in revenue and profits will be possibly be lost [66]. Previous and current research of data analytical methods showed to be functional on huge amounts of unanalysed health and medical patient data, and their results can provide more information on patients. Furthermore, discrete and population data can notify the patients and their doctors when a decision is being made and it can help decide the most suitable treatment options for these patients. To discover the current methods for data collection in hospitals, a review of relevant literature from medical journals and conference papers from specialist sources such as the British Medical journal (BMJ) must be carried out. During discussions with members of the Alder Hey Innovation Hub, it was confirmed that the NHS currently identifies most of its feedback based on a patient experience survey along with the friends and family test. However, current literature suggests that the use of the friends and family test is limited due to the small range of patients that wilfully complete the form, who are usually of negative opinion [88]. Other methods include patient panels, focus groups and the mystery shopper data collection style often used within retail. All of these methods require lengthy processing times [2], which increases the action time between collation of feedback and action being taken. Any delay between feedback and action in a hospital setting can result in patients feeling as if they are not being listened to. In fact, immediate feedback can an informed and involved patient, which not only does improve health care quality but overall outcomes in care [101]. From a review of published sources, it is apparent that much of the available literature is of a non-experimental, descriptive type that reports case studies of locally implemented systems. Such a trend is not uncommon for research questions that address the effectiveness of organisational processes, data gather-

ing questions posed form managerial positions differs greatly from those posed by clinicians, which also differ from those asked by patients. This issue raised by many reports creates a divide between perceived good health care from a patients point of view and poor service provided by a clinician and good service as seen from a clinician but perceived poor service if viewed from a patient [74]. It is clear that only a small amount of literature exists that directly addresses the action feedback loop in a hospital context, that is the feedback process by which incident data in any form are transformed into beneficial improvements in operational safety or actionable events that produce positive improvements to patients' experiences. An academic review of patient satisfaction surveys reveals that measuring patient satisfaction has numerous problems, as most surveys oversimplify complex issues that each patient faces during his or her hospital stay. Furthermore, the notion of 'satisfaction' is highly debated, which is defined as a judgement people form over time as they reflect on their experience. This experience can differ from quality health care as perceived by a health care professional [37]. Most surveys or feedback gathering exercises have a long processing and administration overhead, leading to lengthy delays between comments, suggestions and experience reviews from a patient transforming into visible action, usually long after the patient has left the hospital. A report from [25] discusses a qualitative interview study concerning the effectiveness of data feedback in supporting performance improvement efforts in eight US hospitals. Data quality, timeliness and credibility were identified as important factors for effective improvement, along with leadership and persistence in data feedback processes "data feedback must persist to sustain improved performance. Embedded in several themes was the view that the effectiveness of data feedback depends, not only on the quality and timeliness of the data but also on the "organisational context in which such efforts are implemented" [25]. To maximise data feedback [25] outlines the importance of the quality, timeliness and context of the data. [61] highlights two critical determinants for success:

1. timely, effective feedback
2. demonstrable utility.

Timely, effective feedback assures reporters that their reports are acted upon and are not trapped into an administrative "Blackhole". Demonstrating the local usefulness of incident data, in addition to the development of external reports, influences user adoption and compliance, and can improve reporting rates. These factors highlight the need for immediate analysis and processing of patient feedback, clinical and hospital data. The ability to follow the action in a timely manner is further highlighted by [47], who discuss the importance within US health care of follow-up actions after a safety/error report. They suggest that more emphasis needs to be placed on the event follow up prioritising opportunities and actions, assigning responsibility and accountability whilst producing an action plan to meet the needs of the reported issue [47]. Current literature treats feedback from patients as a separate category from that produced by clinical staff [67], with differing procedures and protocols in place for dealing with feedback each type. Staff training literature given to NHS staff contains no guidance on the expected turnaround for feedback that is received. Each hospital manages and addresses its feedback, comments and complaints uniquely, with some publishing reports after an unspecified period in some cases this can take upwards of four months. Reports indicate that formal language and taxonomy varies between the hospital setting and background of each patient who provides feedback. A standardised taxonomy to focus on specific areas is recommended to narrow feedback into actionable groups [54]. Creating manageable action groups with patient satisfaction metrics applied to each area. The ability to measure the patient experience is discussed by [69]. It concludes that the most effective method of measuring patient experience is to utilise a mixed-method approach, combining surveys and more narrative data collection methods such as patient stories ensuring the focus remains on what matters most to the patients. The literature relating to hospital data collection indicates that current methods are largely ineffective at providing immediate feedback to patents allowing the patient to feed involved or that any reports they submit will have any impact upon the conditions experienced throughout the hospital. With real-time data collection and feedback opportunities presented by the new data collection method proposed by the Innovations Hub at Alder Hey Chil-

dren's Hospital, the use of free form text entries as well as emotional and personal wellbeing index scores would ensure the data being collected will directly relate to the current experience. This will allow analysis to take place that can highlight issues to be addressed during the patient's current journey.

An industry that currently makes use of 'Big-data' analytics applied to customers' feedback is the retail industry, with most large retail chains utilising a multi-data source model for analysing store performance. This analysis allows the generation of predictions on what aspect of the shopping process has the greatest impact on shopper satisfaction so that shoppers will ultimately spend more money on each visit [46]. This usage of Big Data for simulations technology is designed to allow the evaluation of procedures and current methodologies. Such a method could be modified and implemented within the hospital setting to provide detailed performance metrics concerning patient satisfaction and experience. The data processed is not necessarily required to be real-time, but a fast turnaround on data collection and action being taken is significantly less than those experienced in a hospital. This allows stores to adapt given a consumer change, such methods could prove useful for understanding patient feedback and how to implement rapid change based upon feedback and analysis.

2.1.1 Natural language processing

Natural language processing (NLP) is widely studied and implemented across many disciplines to provide automated processing of human language. The focus of this review is to discover the use of NLP for large data analysis in a medical situation.

Natural language processing is the use of computer systems to process and analyse standard natural user language. Modern natural language processing is widely discussed in many contexts, with significant research effort around the use of NLP for high-level tasks such as sentiment analysis and categorisation of text to provide a detailed analysis of a large data set [83].

NLP is commonly used to examine large free from text corpora, to extract, iden-

tify and assess information relevant to a specific context. There is extensive research focusing on the overall sentiment expressed by customer reviews [5]. Research by Ponsignon [92] demonstrates the efficacy of content analysis methods to identify patterns in the data to assess the overall trends in patients' satisfaction. Furthermore, quality healthcare must be defined individually for each establishment due to the disparity in medical treatment, and invasive procedures as the best option may not appear to the patient as quality. Studies such as [73] have analysed large amounts of data acquired from NHS hospitals in the United Kingdom, which have manually applied analysis techniques to specific datasets. On the other hand, limited research has focused on the integration of machine learning with NLP techniques and [50] suggest they lead to accurate and meaningful results when predicting hospital cleanliness (81% accuracy), treatment with dignity (83% accuracy) and overall recommendation (89% accuracy). These results have been produced using standard machine learning and NLP software Weka, with training data from NHS choices website comments from 2008, 2009 and 2010. More specifically, custom prior classification of 1000 most common words and phrases contribute to enhanced accuracy levels. The report suggests that machine learning and NLP techniques lead to a valuable analysis and monitoring of patients' experience in real-time. These findings have been replicated and a similar conclusion has been reached by [77]. However, these studies focused on freely available and generic solutions.

The above-mentioned work demonstrates the feasibility of using machine learning and NLP to create meaningful solutions addressing hospital data. With the direct involvement of Alder Hey Innovation Hub, bespoke machine-learning algorithms and suitable NLP processing techniques will be designed and implemented, to create a meaningful output, which would be utilised as a part of their internal initiative to harness patient's feedback to provide enhanced, efficient and successful healthcare system. With such a wide range of models available and each providing a unique solution to NLP problems, the correct framework must be selected to provide an appropriate solution to the different scenarios to correctly predict and analyse the textual input. Integrated use of supervised and unsupervised machine

learning [14], should be used, allowing a better analysis of large data sets. This would also be utilised to train suitable machine learning techniques to ensure the implementation of the most accurate categorisation.

2.1.2 Lexical analysis

The most basic level in an NLP system is based on lexical analysis, which deals with words regarded as the atomic structure of text documents; in particular, it is the process which takes place when the basic components of a text are analysed and grouped into tokens which are sequences of characters with a collective meaning. In other words, lexical analysis facilitates the interpretation of individual words, which can refer to more than one concept, based on the context in which they occur. As a result, the use of simplified lexical representations unify the meaning across words to generate complex interpretations at a higher meta-level. The lexical analysis may require a lexicon, which usually consists of the particular approach used in a suitably defined NLP system, as well as the nature and extent of information inherent to the lexicon. Mainly, lexicons may vary in terms of their complexity as they can contain information on the semantic information related to a word. Moreover accurate and comprehensive sub-categorisation lexicons are extremely important for the development of parsing technology as well as vital for any NLP an application which relies on the structure of information related to predicate-argument structure. More research is currently being carried out to provide better tools for analysing words in semantic contexts.
More specifically, the lexical analysis consists of various tasks, which include

- Lemmatisation, which collects inflected forms of a word into a single item corresponding to its lemma (or dictionary form)
- Part-of-speech tagging, which aims to identify the syntactic role of each word
- Parsing, which is the process to grammatically analyse a sentence, where the contribution of each word is considered as a whole, with the corresponding

hierarchy

2.1.3 Semantic analysis

The semantic analysis deals with a higher meta-level with respect to the objects associated with a lexicon. Semantic processing determines the possible meanings of a sentence by investigating the interactions among word-level meanings in the sentence. This approach can also incorporate the semantic disambiguation of words with multiple senses. Semantic disambiguation allows selecting the sense of ambiguous words, given a context. So that they can be included in the appropriate semantic representation of the sentence, for example, 'The trophy would not fit in the brown suitcase because it was too big', A human who reads this sentence can identify the 'it' regarding the trophy by using their knowledge about the typical size of objects and their ability to do spatial reasoning can decide it was the trophy that is too big for the suitcase, a series of challenges including this sentence is taken from the Winograd schema [71], this schema has been developed and used to evaluate semantic disambiguation. The English language contains a wealth of ambiguous words that without disambiguation can cause issues for NLP. This is particularly relevant in any information retrieval and processing system based on ambiguous and partially known knowledge as mentioned above. An interesting aspect of this research field is concerned with the purposeful use of language, where the utilisation of a context within the text is exploited to explain how extra meaning is part of some documents without actually being constructed in them. This is still being developed as it requires an incredibly wide knowledge dealing with intentions, plans, and objectives. Extremely useful applications in NLP can be seen in inferencing techniques where extra information derived from a wider context successfully addresses statistical properties.

2.2 Using Social media to Improve Hospital services

[127] Outlines significant usage of social media within healthcare to promote patient interaction and to disseminate good health care practices such as health ad-

vice. The report highlights how social platforms such as YouTube are becoming more popular as doctors and health professionals are ale to interact with patients and provides advice and guidance on common health issues. The heath care professionals are also using social media to interact with other professionals and organise Continual professional development and to encourage collaborations between hospitals. The use of social media has improved patients opinion and use of facilities demonstrating an advantage over not utilising social platforms.

2.3 Existing Technology

Data has become a crucial part of most of the scientific fields, as well as business, social sciences, humanities and the financial sector. Data is continuously created by capturing human activity, financial transactions, sensor information, to name but a few. Therefore, the ability to identify actionable insights and useful trends has become a priority for many organisations, especially when applied to multidisciplinary contexts [79].
Big Data research mainly focuses on four main properties, although in a different context a higher number of such properties are considered [51], namely:

- **Volume:** the amount of data produced daily is enormous. The combination of real-time and historical data provides a wealth of information to facilitate the appropriate and best decision process.

- **Velocity:** real-time data raises numerous challenges as suitable processing power must be allocated to allow an efficient assessment within the specific time constraints. However, depending on the sources, type and dynamics of such data, various techniques need to be implemented to provide sufficient efficiency.

- **Variety:** data consists of various types, structures, and format. For example, information is collected from audio or video sources, as well as from sensors and textual sources, to name but a few. This diversity requires suit-

able tools and techniques that can be applied to efficiently deal with different data types.

- **Veracity:** data is likely to contain contradictory and erroneous information, which could jeopardise the whole process of acquisition, assessment, and management of information.

A selection of existing solutions have been evaluated from high profile industry leaders.

Alteryx [[8]] offers a unified machine-learning platform, which facilitates the design of models in a single workflow. With the combination of machine learning algorithms and some visualisation techniques, the Alteryx solution attempts to unify all aspects of data analytics. In particular, it focuses on business users and data scientists, and it attempts to address the skills shortage in this field, by enabling the creation and execution of models intuitively.
However, Alteryx is regarded as only a data preparation solution vendor. Whilst the software does include some other features for the reporting and visualisation of data, they remain comparatively weak with respect to the intuitiveness of the rest of the platform. This area is a large downfall to the overall solution. A lack of overall support and compatibility with Unix systems creates would limit enterprise roll out to Windows-based organisations.

The Anaconda Suite [[9]] is an open-source development environment based on the Python and R programming languages. It is a centralised repository for open source libraries to reside and be continually upgraded to ensure the latest version is available to all users. The system enables real-time processing and live data analytics with the use of a Python, R-notebook or live environment (Jupyter). Whilst a large portion of libraries are well maintained and have been carefully developed, it requires users to report fallacious or malicious libraries as the Anaconda developers have limited control over the quality and reliability of the code. The open-source nature of Python and R allows developers the autonomy to fully customise and tailor the approach taken to maximise the output so that it aligns with

the business aims and objectives. The approach taken by the Anaconda team is to create a platform that can be freely customised. If a developer decides additional functionality is required, it is entirely possible to add such functionality within the Anaconda framework with additional Python code. The machine learning capabilities of Python are far-reaching, as it is currently one of the most popular choices for creating machine learning solutions.
However, every component provided by Anaconda must be custom developed. As a consequence, extensive Python or R knowledge is required to developed and implement a solution within the Anaconda system.

IBM [[58]] offers a wide range of machine learning and data analytics solutions, with over 40 different products available, each focusing on small analytics or machine learning solutions. The text analysis and data mining solution SPSS and Watson allow the identification of business-ready results, which makes it a popular choice across the business world, with IBM currently possessing the market share of users at around 9%. Their intuitiveness enables all users to harness the machine learning processes with the data preparation and model management aspects.
However, a large number of solutions provided by IBM propose a fragmented set of the machine learning options, which can create confusions as to what exact learning algorithm and output would be most suitable to a specific business application. Furthermore, the majority of the solutions are single-use applications with no ability to tailor the software and its interface (as expected from closed source software).

Microsoft [[78]] much like IBM, provides a wide range of products suited and aimed at the machine learning and data analytics community. The use of Microsoft Azure cloud services allows a connected system that provides a scalable cloud solution. The two major routes Microsoft offers include Azure Machine learning (Studio) and its onsite solution of SQL Server with Machine Learning Services. The overall solution offered is limited in its advanced options but does provide a suitable introduction to Machine Learning for business intelligence. Being one of the

most recognisable names in technology Microsoft has substantial market awareness and capital to create a product that meets the consumers' needs. However, the ability to provide a cloud-based solution enables smaller business to enjoy the benefits of computationally expensive machine learning at a fraction of the cost. This solution requires trust in the cloud services to maintain data confidentiality with potentially significant risks. The lack of ability to customise the application to address specific business needs can also cause problems if the domain is highly focused. The machine learning studio offering lacks in some critical areas, including code synthesis, containerisation and external code access. Users are also unable to maintain a stable working version. If Microsoft pushes an update to the services users can't retain the current version.

The University of Waikato, New Zealand, created two software solutions: Weka [56] with a primary focus on regression and classification tools, and MOA [21] for streaming data analysis. These packages are both free (GNU licensed) and they contain a collection of visualisation tools and algorithms for data analysis and predictive modelling. Its simple GUI provides access to all its functionality with no coding knowledge or specialist training. The machine learning functions are implemented in Java and simple data pre-processing is also available. The tool originated with the processing of agricultural data but has since been developed in a full machine learning-based analysis platform, capable of running across any system capable of running a Java executable. The Weka and MOA solutions allow for simple analysis and models to be built with no complications, which is the main reason for its popularity. The system is based on a "plug and play" model aimed at entry-level machine learning and data analysis. In comparison to the other methods available, Weka is significantly less flexible for statistical analysis and data exploration. Unlike its competitors, it tends to be difficult to modify and clean datasets. Furthermore, the package cannot explore and transform data sets without fully understanding the source code and altering the application and the packages source code requires a high level of expertise.

2.4 Natural Language Processing in Big Data

Natural Language Processing (NLP) consists of a range of computational techniques for assessing and extracting knowledge from textual sources, via linguistic analysis for a range of tasks or applications. The goal of NLP is to extend its methods to incorporate any language, mode or genre used by humans to interact with one another, to achieve a better understanding of the information patterns that emerge in human communication. NLP was originally referred to as Natural Language Understanding (NLU) and even though the ultimate target of NLP is "true"' NLU, there is still much research required to achieve that. The ability to logically infer conclusions from textual sources is still being developed and improved to incorporate the richness of language in terms of imprecise knowledge, causality and ambiguous meaning.

2.4.1 Information extraction via NLP techniques within big data

One of the most investigated data types in Big Data includes those without a well-defined structure, or unstructured. Furthermore, text mining techniques allow the identification and assessment of relevant information from textual data sources [34].

Depending on specific semantic properties, different text mining techniques can be utilised.

One aspect of NLP focuses on sentiment analysis, which aims to detect "opinions" or *polarity* from textual data sources [72, 95]. In [122], the authors define an extensive set of keywords and cue phrases generated by automatically extracting them from the tagged version of the Brown Corpus, which contains approximately 500 samples of English-language texts [44]. This was carried out by considering the triples `(NP1, VB, NP2)` where

- `NP1` and `NP2` are the *noun phrases*, i.e. phrases with a noun as its head word [43], which had to contain one or more specific keywords (e.g. Rapid heartbeat, blurry vision.)

- VB is the *linking verb*.

The NP1 and NP2 were assessed to identify appropriate keywords, and cue phrases. The main steps included the following:

- The Stanford Parser (A typed dependency parser of English sentences from phrase structure parses with labelling of grammatical relations) [34] was used to shallow parse (identification of constituent parts within a sentence including nouns, verbs and adjectives etc.) textual fragments from input datasets.

- A grammar-based extraction identified the triples `(NP, verb, keyword)`, where NP, is the noun phrase, `verb` is the linking verb, and `keyword` consists of one or more keywords as mentioned above.

Such triples were utilised to define the nodes and edges of a network, by identifying any connection among the keywords defined above.

2.5 Network Theory

A critical component in the data processing, which will take place during this project is the creation of networks to model the relationships between the different components embedded in the extracted information. Complex networks model a wide range of systems and scenarios, with applications to biology, sociology, psychology and the internet [7]. In particular, utilising network models allow the discovery of patterns and properties of complex systems, in addition to the interconnections of their different sub-components. Moreover, the availability of accessible computing power allows the processing of large networks to determine their associated network in a more efficient manner. Current modelling approaches have demonstrated that real-world networks tend to exhibit non-random behaviour as they follow specific scale-free and small-world organising principles that are shared by several complex systems [7]. This approach will be applied to

the hospital data extracted during the project, which will create suitable networks capturing all aspects of the hospital patient data, clinical information and hospital usage information. This will potentially allow a deeper information discovery as well as the ability to predict influence across the entire range of data. Work regarding the structure and function of complex networks undertaken by [85], further supports the fact that the investigation of network theory can be applied to broader and larger datasets. Network analysis previously focused on small graphs and the properties of individual vertices. However, modern approaches have been investigating the use of a variety of computational approaches to huge networks consisting of millions of nodes and mutual links. New research directions now focus on what components of such networks play a significant role in the overall behaviour of the modelled scenario. Regarding the aim of this project, no single sub-components of the network associated with the information extracted from hospital data is likely to be the sole contributor to the overall system. The ability to populate the appropriate network will lead to a deeper understanding of the overall organisational behaviour of the data and its impact upon other areas related to the hospital. Furthermore, this would allow the investigation of the dynamical properties of the information, which would contribute to the identification of the crucial aspect with a significant impact on patients' journey.

Network theory has become increasingly popular in numerous research fields, including mathematics, computer science, biology, and the social sciences [130]. More specifically, networks are defined as sets of nodes $V = \{v_i\}_{i=1}^{n}$, which are connected as specified by the edge-set $E = \{e_{ij}\}_{i \neq j=1}^{n}$ [16]. Real-world networks are utilised to model complex systems, which often consist of numerous components. Therefore, the resulting complexity can lead to models, which are computationally demanding. To balance accuracy with efficiency, in [122], [121] and [118], the authors propose a method, based on data and text mining techniques, to determine and assess the optimal topological reduction approximating specific real-world datasets. In [120], the topological properties of such networks are further analysed to identify the connecting paths, which are sequences of adjacent edges. This approach enables the identification of the mutual influences of any

two concepts corresponding to specific nodes.

The importance of such a process is that it allows the identification of a topological structure which can give an insight into the corresponding data-sets. It is possible to extract information on the system modelled by such network that can be used to determine relevant intelligence. The algorithms utilised for the reduced network topology extraction process are introduced, and the reader can refer to that article for further details. Furthermore, these algorithms also allow the identification of the long-tail distribution in the case of scale-free networks, resulting in a more accurate and relevant extraction [118, 122].

Another area based on Network Theory, which has been attracting considerable attention is Human Dynamics [35, 117]. Human activity usually exhibits complex properties, which are analysed by using a variety of models based on the dynamical properties of the associated systems, as well as the topology of the corresponding networks generated by mutual interactions.

The rest of this section will focus on an overview of random and scale-free networks, which are used to topologically reduce real-world networks [120].

2.5.1 Random Networks

Random networks are defined by probabilistic processes, which govern their overall topology and the existence of any edge is based on a probability p. Such networks have been extensively investigated, and several associated properties have been identified depending on their theoretical, or applied context. More specifically, the fraction p_k of nodes with degree k is characterised by the following equation

$$p_k \approx \frac{z^k e^{-z}}{k!},$$

where $z = (n-1)p$ [16].

When random networks are used to model real-world scenarios, then the relationships among the nodes, are purely random. In such a case, The edges connecting nodes, the relationships captured by the edges are unlikely to be associated with meaningful influence. If a random network is associated with a purely randomised

system, then the relations between nodes do not follow a specific law [120]

2.5.2 Scale-Free Networks

Scale-free networks appear in numerous contexts, including the World Wide Web links, biological and social networks [16], and the continuous enhancement of data analysis tools is leading to the identification of more examples of such networks.
These are characterised by a node degree distribution, which follows a power law. In particular, for large values of k, the fraction p_k of nodes in the network having degree k, is defined as

$$p_k \approx k^{-\gamma} \tag{2.1}$$

where γ has been empirically shown to be typically in the range $2 < \gamma < 3$ [16].

A consequence of Equation 2.1, is the likelihood of the existence of highly connected hubs, which suggest that in scale-free networks the way information spreads across them tends to exhibit a preferential behaviour [16]. Another important property is when new nodes are created, these are likely to be connected to existing nodes that are already well linked. Furthermore, since the connectivity of nodes follows a distribution which is not purely random, networks that are topologically reduced to scale-free structures are likely to capture influence relations between the corresponding nodes, and their dynamics provide predictive capabilities related to their evolution.

2.5.3 Topological Data Analysis

Topological Data Analysis (TDA) is an emerging research area, which combines methods from network theory with topology to classify and analyse complex data [27]. The aim of TDA is the assessment of the structure of the corresponding data, defined by the connectivity of its components. In particular, invariant features of a data-space, or in other words general properties that do not change, play a crucial role in their classification. For example, stretching an elastic band will not change the fact it has a hole. In fact, in several clustering and classification methods, in-

variant properties of objects provide valuable tools, as different types of data connectivity can be utilised to group similar data clusters together. Two elastic bands might have different dimensions, yet they are still considered similar due to their "hole". This is captured by the concept of *persistent homology*, which focuses on the identification of the topological properties which remain invariant [40].
One of the building bricks of persistent topology is simplicial complexes, which are space triangulations defined by combined, non-overlapping polyhedra, covering a topological space. These include Voronoi diagrams, Delaunay triangulations, Vietoris and Čech complexes. An approximation of an image by a suitable pixelation, aiming to provide an accurate representation, is an example of space triangulation. In fact, one of the most important aspects of simplicial complexes is that fact that they provide an "approximation" of the corresponding objects. There are several TDA implementations for different needs and contexts, which have been shown to produce good results. For an overview, refer to [95].

2.6 Influence and Causality

Big Data is increasingly influencing the way we obtain, assess, and manage information [79]. In particular, a very powerful and efficient approach to obtaining insight into real-world Big Data is by determining the main properties that characterise such data-sets, and the factors that influence them. Influence and causality between concepts are an important issue within Big Data research. The main difference between them is often semantic, as the sentence *"chemotherapy influences an improved outlook of certain types of breast cancer"* is profoundly different from *"chemotherapy causes an improved outlook of certain types of breast cancer"*. Causality is a much stronger statement based on stricter conditions compared to influence, and the former implies the latter, but not the other way round [89]. In other words, causality allows a more direct link between concepts, which enables a more conclusive and well-defined decisional approach [119]. Furthermore, causality implies a direction (*"A causes B"* is different from *"B causes A"*), and it is often characterised by semantic unambiguity.

As discussed above, influence describes a weaker, and more general concept compared to causality, particularly from a semantic point of view. In fact, influence between two or more concepts may not be tied to a direction, or a well-defined, unambiguous semantic definition. There is extensive research on the automated extraction of causal relations between concepts, such as events, entities, factual data, etc. [23]. In many contexts, influence between two objects is often considered as based on of their mutual co-occurrence. However, co-occurrence does not necessarily imply any influence, since their mutual existence might be completely unrelated. When analysing large amounts of data, the co-occurrence of two or more elements can facilitate the extraction of specific insights. For example, this can allow large datasets to be topologically reduced to determine whether the data follow a scale-free or a purely random structure [122].

The network structure provided via semantic analysis in NLP provides a tool for modelling stochastic processes within complex systems. In [114] some properties of semantic networks are demonstrated to have important applications to the processes of semantic growth. Such properties are based on statistical properties linked to theoretical properties of the associated semantic networks. Furthermore, such networks exhibit small-world structures characterised by highly clustered neighbourhoods and a short average path length [130]. Such networks also show a scale-free organisation [16] defined by a relatively small number of well-connected nodes, with the distribution of node connectivities, which is governed by a power function.

Causal inference is one stage of a crucial reasoning process [30, 42] which plays a fundamental role in any question-answering technique with interesting AI applications such as decision-making and diagnosis in Bayesian Networks (BNs). Moreover, they provide a powerful tool in achieving knowledge about causal rules. In particular, the investigation on how to obtain them is the crucial step in developing systems capable of causal inference especially in complex domains [19]. The conditional dependencies in a Bayesian Network are often based on known statistical and computational techniques and one of their strengths is the combination of methods from graph theory, probability theory, computer science, and statis-

tics. Such networks heavily rely on data and providing them with text information is potentially a very powerful way to analyse causal relations. In [104] an investigation on data acquisition from text to generate Bayesian networks is carried out.

Causal learning often focuses on long-run predictions through an estimation of the parameters of a causal Bayes network structural learning. An interesting approach is described in [33] where people's short-run behaviour is modelled through a dynamical version of the current approaches. Moreover, the limitation of a merely static investigation, or in other words the judgements are made after observing all of the data is addressed by a dynamical approach based on BNs methods. Their result only applies to a particular scenario but it offers a new perspective and it shows huge research potential in this area.

Often, any two concepts linked by paths in the semantic network they are embedded in, are difficult to understand in terms of their influence. As a consequence, either the topological structure of the network is not fully known or there is partial knowledge of the structure of the paths between them. An important concept to understand the influence between two concepts is *causality discovery* [42] which aims to pinpoint the causal relationship between them. Typically, semantic similarity measurement plays a significant role in semantic and information retrieval in contexts where detection of conceptually close but not identical entities is essential. Similarity measurement is often carried out by comparing common and different features such as parts, attributes and functions, in [59] a method based on adding thematic roles as an additional type of features to be compared, and it describes the reason why the use of thematic roles may prevent wrong function matches. Semantic distance is closely linked to the causal relationship as it describes how *closely* two concepts are connected. However, much of the work on this topic is concerned with the linguistic or semantic similarity of terms based on both the context and the lexicographic properties of words. One of the main setbacks of this approach is that a hierarchical structure of the concepts can lead to an oversimplification of the problem. The important question is not merely how far two concepts are, but *how much a concept is influential with respect to another one*. The difference is subtle but crucial when dealing with causal discovery. The se-

mantic distance can also be applied to information retrieval methods to improve automated assignment of indexing based descriptors, as well as to semantic vocabulary integration which enables to choose the closest related concepts while translating in and out of the multiple vocabularies.

2.6.1 Bayesian Networks in Big Data

The importance of Bayesian networks within statistics and machine learning are crucial in particular an understanding of Bayesian networks are required to understand bayesian inference (updating the probability of a hypothesis as more information becomes available) and Bayesian causal inference (BCI). Bayesian Networks (BNs) [60, 89] are very powerful tools with a wide range of applications with particular emphasis on cause and effect modelling in a wide variety of domains. Loosely speaking their main characteristic is the ability to capture the probabilistic relationship between variables, as well as historical information about their relationships. More formally, Bayesian networks are directed acyclic graphs such that their nodes represent Bayesian random variables or in other words, they are associated with observable quantities, unknown parameters, hypotheses, etc. Nodes that are conditionally dependent are joined by an edge. BNs have proved to be very successful when a scenario consisting of already known information coupled with uncertain or partially known data, is considered. Moreover, such networks also offer consistent semantics to described causes and effects via an intuitive graphical representation and we will discuss this in Section 2.6.

Our knowledge and representation of the world are usually based on unknown parameters from the uncertainty of *a priori* knowledge and partial or total lack of certainty of a particular scenario. BNs provide a tool to model uncertainty and intuitive graphical representation of the interactions between various events, generating a powerful method of modelling cause and effect scenarios. A BN represents the possible states of a defined domain-containing probabilistic relationships among some of the states. Conditional probability tables describe the likelihood of any node in the Bayesian network being in one state or another without current

evidence and in particular, they depend on the causality relationships between some nodes often described by prior information on such networks.

2.6.2 The Bayes' Rule

Bayesian networks are based on the conditional probability theory developed by Thomas Bayes [60, 89], who discovered a basic law of probability also called *Bayes' rule*, namely

$$P(b|a) = \frac{P(a|b)P(b)}{P(a)}, \tag{2.2}$$

where $P(a)$ and $P(b)$ are the probability of a and b respectively and $P(a|b)$ is the probability of a given that b has occurred. Equation 2.2 can be also expressed in more general and complex terms if we consider the scenario where we can update our belief given by hypothesis H according to additional evidence E and the past experience c so that we have

$$P(H|E,c) = \frac{P(H|c)P(E|H,c)}{P(E|c)}, \tag{2.3}$$

where the above terms are referred to as follows [89]

- $P(H|E,c)$ is defined as the *posterior probability* of H based on the effect of the evidence E.
- $P(H|c)$ is the *a priori probability* of H based on c.
- $P(E|H,c)$ is the probability of the evidence E following the assumption that the hypothesis H and c are true.
- Finally $P(E|c)$ is independent of H and is often defined as a normalising or scaling factor.

As an example, Table 2.6.1 from [86] shows the probability of rain using some fixed probabilities.

Table 2.6.1: Example of marginal and joint probabilities for rain both today and tomorrow [86].

	Rain Tomorrow	No Rain Tomorrow	Marginal Probability of Rain Today
Rain Today	0.14	0.06	0.20
No Rain Today	0.16	0.64	0.80
Marginal Probability of Rain Tomorrow	0.3	0.7	

The use of Bayesian networks in NLP which has proved to be a research area with huge potential as the ability to deal with probabilistic knowledge is a clear advantage of such approach.
Automated NLP understanding systems rely on several data sources which are often partially or very little known, resulting in problematic tasks as the integration of disambiguation and consequently the use of probabilistic tools has proved to be very challenging. Even though such knowledge sources are well known to be probabilistic with well-defined models of some specific linguistic levels, the combination of the probabilistic knowledge sources are still little-understood [84]. Bayesian networks applications to NLP have clear advantages and in particular:

- Quantitatively evaluates the impact of different independence assumptions in a uniform framework.
- The possibility of modelling the behaviour of highly structured linguistic knowledge sources with local conditional probability tables as well as the use of well-known algorithms to update the Bayesian network which can evaluate the global influence of new evidence [60].
- The exploitation of on-line interpretation algorithms, where partial inputs correspond to partial evidence on the network so that different nodes are instantiated and the posterior probability of different constructions changes appropriately.

In NLP, the relationship between subject and verb can create a variety of problems. The prediction of phrase structure trees for sentences may cause processing difficulties as there is a vague probabilistic semantics for the induced hidden representations. In [55], Incremental Sigmoid Belief Networks are investigated which are graphical models similar to a class of a neural network with a clear probabilistic semantics for all their variables described by using BN algorithms.
The ambiguity of the syntax and semantics of natural language makes the development of rule-based approaches very challenging to address even very limited domains of text. This has led to probabilistic approaches where models of natural language are learnt from large text sets. A probabilistic model of a natural language subtask consists of a set of random values with certain probabilities, associated with lexical, syntactic, semantic, and discourse features [89] and the use of Bayesian networks applied to multiple natural language processing subtasks in a single model supports inferencing mechanisms which improve simple classification techniques [90]. Task classification is an important part of BNs [89] and a successful method is naïve Bayes approach due to its algorithmic efficiency and its relative simplicity in training and running, and it is well-established for classification tasks, in [70] the content and accessibility, as well as readability in medically related texts are discussed by using a vocabulary based on naïve Bayes classifier which is analysed to distinguish between difficulty levels in text and to investigate the challenges that people face when searching for information to maintain or improve their health. Moreover in [87] the authors discuss ClearTK which is a comprehensive suite of tests for a statistical natural language processing toolkit. Mainly it provides a framework to develop UIMA analysis engines based on statistical learning as the base for decision making and annotation creation.

2.6.3 Extraction of Bayesian Networks from Text

Usually, BNs are analysed, defined and built out manually by expert modellers. However, this is a very time-consuming process and only a small amount of data sources can be analysed [89]. To address these issues, there has been extensive

research on the extraction of BNs from unstructured data, with particular emphasis to textual sources [104]. However, this is a complex task due to the intrinsic ambiguity of natural language, as well as to the strict topological and probabilistic rules, which BNs need to obey. In particular, challenges associated with low recall and precision, as well as contradictory information, must be addressed to provide a reliable BN automated extraction tool.

In [116], the authors introduce a method to extract and populate fragments of BNs from biomedical textual sources, defined on grammar and lexical properties, as well as on the topological features of the associated networks. More specifically, a text pattern approach was utilised to identify specific concepts and their mutual relations captured via text patterns. This was carried out by considering the following quintuples `(NP1, MOD, tense, keyword, NP2)` where:

- `NP1` and `NP2` are the *noun phrases*, or in other words phrases with a noun as the headword, containing biomedical concepts.
- `keyword` refers to probabilistic terms contained in an ontology.
- `MOD` is the modality `keyword`. More specifically, this can be either `positive` or `negative` depending on whether it supports the existence of a probabilistic relationship.
- Finally, `tense` refers to the tense of the verb, which can be either `active` or `passive`. If it cannot be determined, then it is defined as `unknown`.

Subsequently, the network generated by the concepts and relations extracted above is analysed to identify its topological properties, which lead to the most appropriate BNs related to specific term-queries. The evaluation results demonstrate the potential of this approach, especially in providing valuable resources to BN modellers.

2.7 Inconsistencies in Big Data Analysis

A key area in big data analysis is the investigation and resolution of inconsistencies in data [129]. Closely related to this is finding a way to organise data so that concepts having a similar meaning are related through links, while concepts that are distinct from one another are also represented [15]. One of the important benefits in representing data through their relationships is that it will enable the classification and analysis of real-world networks and this will lead to a better understanding and prediction of the properties of the systems that are modelled by these networks [120]. Furthermore, such links will allow more intelligent and effective processing by query engines and analytic tools.

Different expression or statements exist in datasets, which may include some inconsistencies. However, to extract meaningful links or create accurate and relevant information from structured and unstructured data, it is important that the inconsistencies present in data are identified and addressed. The focus of this section is on the variety and veracity of Big Data. As discussed in Section 2.3, variety refers to different forms of structured and unstructured data sets that are collected for use. Also, data contain erroneous, contradictory and missing information which potentially undermine the whole process of acquisition, assessment, and management of information.

More specifically, data over the real world are unstructured and in various formats. Data inconsistencies could occur during the analysis and integration of data from different sources, where each source may represent the same information differently. Inconsistencies could potentially exist both at the data value level and the data format levels. In fact, in [129] the authors have found that inconsistencies exist at schema level, data representation level and data value level and this is being seen as a data analysis and integration problem. To provide effective information applications, an organisation would typically require data from these multiple sources. However, analysing and integration of such data present their own challenges, with respect to the overall business context. For example, misinterpretation of data resulting from inconsistencies could potentially affect the overall

result of the business context. In order to extract meaningful links or create accurate and relevant information from structured and unstructured data consistently, it is expected that a common conceptual model for integrated data should exist [12]. More specifically, it is expected that data are stored in a consistent format in order to facilitate intelligent analytics. Moreover, the benefits of virtualised access to multiple data sources inconsistent types and formats are enormous. To this extent, data formats must be consistent across the operational context to be acted upon. Therefore, tackling such an issue is at the very core of Big Data science.

The risk map model depicted in Figure 2.7.1 describes the events which characterise the risk of big data inconsistencies.

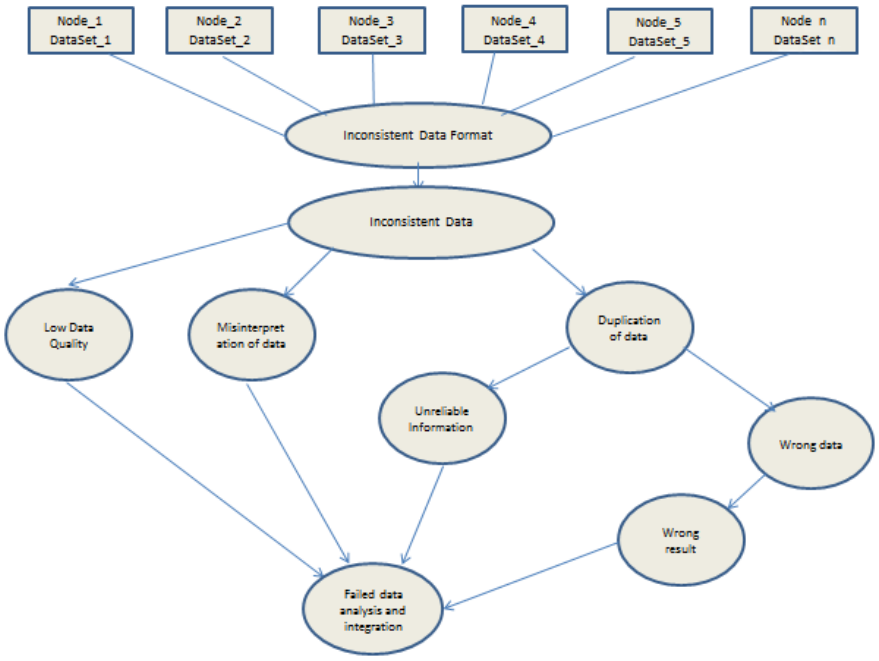

Figure 2.7.1: Risk Map-Dimension of data inconsistencies [98]

2.7.1 Data Set Inconsistencies

Many real-world datasets contain instances of conflicts, inaccuracies and fuzzy duplicates at various levels. In some cases, datasets contain overlapping information

while in other cases; they contain missing information about an entity. Moreover, some contain a different representation of the same real-world entity in the same datasets or different datasets. Inconsistencies exist at the schema level, data representation level and data value level [129]. At the schema level, the presence of different schemas within the same data model gives rise to inconsistencies. This kind of inconsistencies is largely due to the heterogeneity nature of the sources of data. Schema inconsistency is also referred to as Structural conflicts [28]. For example, in an integration scenario where there is no one-to-one correspondence between the tuples of the relations that describe the same real-world entity, one relation could have four attributes, while the second has 6 attributes, a situation commonly referred to as outliers. At the data representation level, inconsistencies arise as a result of data expression that comes in different natural languages and data types as well as the measurement systems. For example, in currency, one source can contain currency expressed in USA dollars, and the other in the British pounds. More specifically, a data value inconsistency exists when two objects obtained from different data sources are identified as representing the same real-world object and some of the values of their corresponding attributes differ. That is, the same real-world entity can be represented in several ways. In [36], the authors distinguish two kinds of data conflicts; uncertainty and contradictions. Uncertainty is a conflict between a non-null value and one or more null values that are all used to describe the same property of a real-world entity. A contradiction is a conflict between two or more different non-null values that are all used to describe the same property of the same entity. Intuitively, contradiction highlights discrepancies in the description of the same event. For example, *"Jane sold a car to John "* and *"John sold a car to Jane"* are incompatible and highly unlikely to occur at the same time and therefore contradictory. Furthermore, uncertainty is caused by missing information such as null values or missing attributes, contradiction is caused by different sources providing different values for the same attribute of a real-world entity [36]. All these give rise to conflicting circumstances which present itself as data inconsistency problem in big data analysis and integration. Figure 2.7.2 shows the architecture of the major phases in the integration and fu-

sion of heterogeneous datasets and the inconsistency levels that are addressed. The

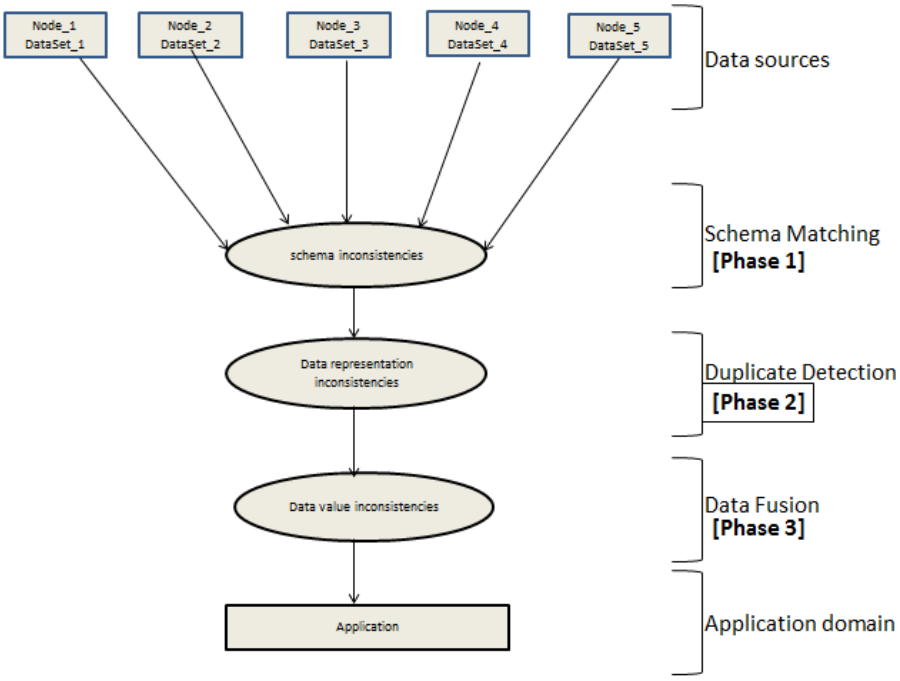

Figure 2.7.2: Architecture of datasets fusion and inconsistency levels [98]

first phase is the schema matching where the schematic mapping between the contents of the respective data sources is done. This is basically the extraction phase in a typical Extract-Transform-Load (ETL) framework. At this phase, schema inconsistencies are identified and resolved. The second phase is the duplicate detection where objects that refer to the same real-world entities are identified and resolved. This is at the tuple level which is basically the transform phase in the ETL process. The final phase is the fusing of data which is the process that involves combining multiple records that represent the same real-world object into a single, consistent state. This is the phase where the process performs attempts to resolves conflicts associated with the datasets. The identification of data value inconsistency is the final state. Therefore, such identification is only possible when both schema inconsistencies and data representation inconsistencies have been resolved. These

kinds of inconsistencies are not universal; rather they are hidden and contextual.

2.7.2 Inconsistencies from Textual Sources

An unstructured dataset lacks defined characteristics and therefore lacks relations which can convey the precise information about the content. An inconsistency in-text occurs when there are conflicting instances of concepts in the same dataset or different datasets from textual sources. That is if two statements in a dataset are referring to the same event or entity. This is known as co-reference which is a necessary condition for text inconsistencies [34]. Another condition is that they must involve the same event and embedded texts must be considered in this determination. This type of inconsistencies affects the integrity of the dataset and can ultimately affect data analysis. For example, *"today is cloudy"* would be perfectly fine and consistent. However, if we also had that *"today is very sunny"* within the same datasets, this would add a level of inconsistency. Also, assume we have the relation 'MemberOF' as asymmetric property, therefore, the triples 'John $\rightarrow$ MemberOF $\rightarrow$ Conservative Party' and 'ConservativeParty $\rightarrow$ MemberOF$\rightarrow$ John' is in a state of inconsistency. These are two instances that represent the same real-world entity. These kinds of inconsistency give rise to conflicting circumstances which present itself as data inconsistency problem in big data analysis and integration.

2.7.3 Temporal and Spatial Inconsistencies

Temporal inconsistencies arise when there is conflicting information between two time-series data. In this situation, some data items with reference to time can overlap or temporally coincide in different datasets or the same data set. In particular, this type of inconsistency is crucial when specific inferences are drawn from elements in datasets, such as causality. Such inconsistencies could be partial or complete [136]. Partial inconsistencies occur when the time intervals of two inconsistent events are partially overlapping while complete inconsistencies situation arises when time intervals of two inconsistent events coincide or satisfy containment.

Spatial inconsistencies arise when there are violations of spatial constraints in a dataset with geometrical properties and various spatial relations. For example, when a spatial object in a data set is having multiple conflicting geometric locations, such inconsistencies can arise. Moreover, it can occur in a data integration project when multiple sources with special dimension are represented to the extent that the aggregation of the object violates some kind of unique constraints [137].

An element in datasets may exhibit different kinds of properties and this largely depends on the source of data. For example, location or time series based data may exhibit some form of spatial-temporal properties. Further inconsistencies could also arise from the aggregation of information from textual data sources. Moreover, unstructured text data may exhibit forms of properties that take a semantic and syntactic dimension pertaining to asymmetric, antonym, mismatched values and contradiction. Contradictions can occur in terms of data values, in terms of semantics and in terms of their structural representation [28]. Therefore, the semantics of the language is important in discussing the problem of text inconsistencies in datasets. Being able to construct semantic rules based on ontologies can be an important step in identifying conflicts. it will enable the assessment and verification of the concepts which are being considered. One of the most important questions in solving this problem is how to weigh the available information within a given dataset and find the best data value among the conflicting values in a dataset. Moreover, how to find the data value efficiently. Another important challenge when dealing with textual data is how to represent the content to apply statistical function.

2.8 Machine Learning

Machine learning can be broadly categorised into three main groups: supervised learning, unsupervised learning and reinforcement learning. Each group has unique strengths and weakness creating no ideal single solution. When attempting to utilise machine learning for problem-solving the first step will be to understand what category of machine learning the problem fits.

Text classification is one of the largest problems in machine learning with researchers investigating methods of supervised and unsupervised learning. The ability correctly identify salient features of the input text whilst retaining contextual information that may alter the meaning of a phrase or linguistic turns of phrase such as sarcasm, that if read ad verbatim has one meaning but humans can decode into the true intention. In other words, the text appears to the machine-learning algorithm exactly as the user has typed it. For a machine learning application to be successful some considerations must be accounted for as to the structure of the input. A highly accurate production-ready system will require a robust set of training data that will include common mistakes made whilst typing to account for the variance of user input. Static file analysis allows the use of nearly any machine-learning algorithm to process the file and the algorithm can focus on accuracy (f-score) with some compromise on the overall processing time is taken. On the other hand, real-time analysis requires an optimised and efficient algorithm to make the best use of the available resources whilst producing a usable result in a short time. This enables a close to real-time usage of the processed data. Modern computer architecture allows for GPU arrays to perform this processing in a highly optimised manner and a streaming line by line output can be achieved with an algorithm. Giving users an immediate sense of how the data is being processed.

2.8.1 Supervised and Unsupervised Learning

The supervised learning methodology is the most common form of machine learning, which aims to solve a target variable [13]. This outcome variable is reached from a given set of predictors or inputs (independent variables). Using these set of variables, a function is generated, which maps inputs to desired outputs. The process requires a quality set of training data, that allows correlation to be drawn between inputs and the desired output. The training process would continue until the model achieves the desired level of accuracy on the training data, then unseen data can be given to the model and accuracy can be measured of the built system. Popular Supervised Learning methodologies include: Support Vector Machines,

Linear and Logistic Regression, Naïve Bayes, Decision Tree, Random Forest, K-Nearest Neighbour (KNN) [13].

The unsupervised learning methodology requires no target variable and is generally utilised when a series of unlabelled data is to be analysed. The most common purpose for unsupervised learning comes from its ability to clustering or groups unlabelled data and discover hidden structures from a data set. Examples of unsupervised learning include K-means for clustering and Apriori for rule or association discovery [13].

2.8.2 Semi-Supervised Learning

The semi-supervised learning methodology is a combination of both supervised and unsupervised learning, which address problems that contain both labelled and unlabeled data. Many real-world problems fall into this area as it can be expensive to utilised experts in a particular area to label an entire data set often only a few examples are labelled or data sets are incomplete but it is in comparison cheap to gather and store large amounts of unlabelled data. The semi-supervised learning utilises methodologies from both unsupervised and supervised learning techniques. The former discovers the data structure and the latter make best guess predictions for the unlabelled data.

2.8.3 Reinforcement Learning

The reinforcements learning methodology uses complex algorithms to take actions based on its current state. It then re-evaluates the outcome to again make a decision based on its new condition. The machine is trained to assess different scenarios, including making specific decisions in a training environment. This allows a trial and error approach until the most appropriate options are identified. Popular reinforcement Learning methods include Markov Decision Process and a neural network-based NEAT (Evolving Neural Networks through Augmenting Topologies) [113].

2.8.4 VADER for Sentiment Analysis

The semi-supervised learning method discussed above has numerous applications to text analysis. For example, the Python VADER sentiment analysis tool assesses the sentiment polarity of each word from social media platforms [57]. Consider the following example:

```
>>> txt = "this is superb!"
>>> s.polarity_scores(txt)
'neg': 0.0,
'neu': 0.313,
'pos': 0.687,
'compound': 0.6588
>>> txt = "this is superb"
>>> s.polarity_scores(txt)
'neg': 0.0,
'neu': 0.328,
'pos': 0.672,
'compound': 0.6249
```

The VADER polarity calculations are completely independent and can benefit from a multiprocessing pool in Python this is achieved simply by invoking the built-in library multiprocessing and allocating a worker process pool or allow a dynamic worker process with the aid of Billiard [1].

2.8.5 Linear Discriminate Analysis (LDA) and Quadratic Discriminate Analysis (QDA)

To perform text classification a wide variety of analysis methods are available, Linear Discriminate Analysis (LDA) and Quadratic Discriminant Analysis (QDA) are two fundamental classifiers used commonly as a dimensionality reduction technique, which has been adapted for use in Machine learning, by using simple prob-

[1] https://github.com/celery/billiard

abilistic models obtained using Bayes rule. The two methods estimate the class priors, the class means and the covariance matrices, in QDA no assumptions are made on the covariance, which leads to the quadratic decision boundary.
By using the Bayes rule, we alter class k to provide maximum conditional probability, namely

$$P(y = k|X) = \frac{P(X|y = k)p(y = k)}{P(X)} = \frac{P(X|y = k)P(y = k)}{\sum_l P(X|y = l) \cdot P(y = l)} \tag{2.4}$$

These classifiers show the ease of computation and multi-class form, the methods require no hyper-parameters tuning thus reducing the complexity of any implementation. The reduced complexity can increase overall performance allowing for a more real-time result to be displayed to the user. Where logistic regression is a classification algorithm that is limited to, only two-class classification problems, Linear Discriminant analysis enables classification of more than two classes. The major drawback of logistic regression can be seen in three categories:

- Binary classifier linear regression is only intended for binary classification problems. Linear regression could be extended for multi-class but performs poorly in this domain.
- Unstable with well-separated classes a Logistic regression can be unstable when the data contain well-separated classes.
- Unstable with little training data and logistic regression can become unstable when only a small set of training data is available to estimate the parameters.

Linear discriminate analysis addresses each of these points and is referred to as the go-to linear method for multi-class classification problems. The Linear discriminate analysis method should even be compared with logistic regression for binary classification due to how effective and efficient Linear discriminate analysis performs [24]. Representing Linear discriminate analysis is a relatively simple and straightforward proves. For each class, the mean and variance are calculated for

any variable x to be used. If multiple variables are needed, then the mean and variance are calculated again but over the multivariate Gaussian. These values are used to form the Linear discriminate analysis model, which is based on some assumptions about the data. The visualised data follows a Gaussian curve. Each attribute has the same variance or in other words, the values of each variable vary by the same amount (on average). The simplified process can be shown using a binary classification boundary, where the mean value mv of each input x for each class c is estimated as described in the following equation:

$$mvc = \frac{1}{nc} \cdot \sum x, \tag{2.5}$$

where mvc is the mean value of x for the class c, and nc is the number of instances with class c. The variance is calculated across all classes as the average squared difference of each value from the mean, that is

$$\sigma^2 = \frac{1}{(n-c)} \cdot \sum (x - mv)^2, \tag{2.6}$$

where σ^2 is the variance across all inputs (x), n is the number of instances, and mv is the mean for input x. Whilst LDA is commonly used to produce decision boundaries it is possible to utilise the same algorithm for supervised dimensionality reduction, In doing so it will produce an output less then the number of classes, with such a strong dimensionality reduction technique this method is only suitable for a multi-class setting.

2.8.6 Latent Dirichlet Allocation

A highly effective method for unsupervised semantic modelling tool is Gensim [99]. Whilst still able to produce polarity for a given input text, the Latent Dirichlet Allocation reduction technique/classifier allows for visualisations of word groups and phrases that have a similar meaning. This allows for text consolidation and summarization of a large body of text. The Gensim library is being used for its algorithm, to facilitate the creation of a Vector space model being the algebraic rep-

resentation of an object, in this case, the textual input. The Gensim library utilises latent semantic indexing (LSI), using singular value decomposition that identifies patterns in the relationships between terms and concepts contained within unstructured text, based upon the principle that words used in the same context tend to have similar meaning that overall aids in the creation of coherent groups generated from the input text.

2.8.7 Artificial Neural Networks

With a large proportion of real-world problems fitting the semi-supervised learning label, Artificial Neural Networks aim to mimic biological neural networks, allowing complex relationships between inputs and outputs to be modelled in a non-linear manner. Artificial neural network, are often used for simple chatbots to generate basic responsive dialogue with an end-user, capable of training with only small amounts of data and using a bag of words function to transform the sentences into arrays. Using matrix multiplication and a sigmoid function to normalise the values and derivate to measure the error rate. The sigmoid activation function (2.7) has previously been the go-to activation function.

$$f(x) = \frac{1}{1 + e^{-\beta x}} \tag{2.7}$$

However, recent advances and experimentations have highlighted some problems with the use of this function. The sigmoid function has four important drawbacks, that is vanishing gradient problem, non zero centred output, easily saturated, and slow convergence. These have a negative impact on the computational performance. The ReLu and leaky ReLu functions address this issue.

The ReLu function (2.8) allows forward and back passes to be performed via a simple if statement, and whilst the standard ReLu will saturate when the input is less than 0, this can be eliminated with the use of a Leaky ReLu.

$$f(x) = \begin{cases} 0 & for \quad x < 0 \\ x & for \quad x \geq 0 \end{cases} \tag{2.8}$$

Leaky ReLu Function (2.9) eliminates the saturation problem that would otherwise still exist in a standard ReLu function.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0.01x & \text{otherwise} \end{cases} \tag{2.9}$$

The use of ReLu functions has overtaken the sigmoid function as the go-to activation function for the training of large neural networks with vastly increased parameters at the same computational cost, this leads to higher capacity nets that often produce higher test set accuracy.

Typically, Artificial Neural Networks that have been traditionally used for natural language problems, tend to be Recurrent Neural Networks, with their intuitive 'left to right' processing. Convolutional Neural Networks have also been successfully utilised. In particular, the use of the simple 'Bag of words' approach is an oversimplification, which produces good results. Studies have shown that Convolutional and Recurrent Neural Networks networks have the capabilities to perform equally as well on a natural language processing task [135]. Any implementation of a neural network should be able to produce workable results, and with modifications to hidden layers and batch size, either method can be optimised to reach peak performance.

2.8.8 Implementation of Topological Data Analysis Algorithms

As discussed in Section 2.5.3, persistent homology is an algebraic method to investigate topological properties of specific objects, by defining suitable triangulations (simplicial complex) are defined in terms of a specific metric of the underlying space [49, 95]. A specific implementation is the Python library Dionysus, which has numerous algorithms to analyse different data objects and types. Even though Dionysus is a Python interpreter, it is computationally efficient and it provides an accurate data representation [80].

Manifold Learning is an algorithm, which visualises data of high dimensionality by identifying specific low dimensional manifold properties and parameters.

Python library Scikit-Learn [1] contains various method to facilitate this type of analysis, such as Locally Linear Embedding, Modified Locally Linear Embedding, Spectral Embedding, Local Tangent Space alignment and $t-$distributed Stochastic Neighbour Embedding, which allows efficient and accurate data analysis.

More specifically, Mapper is a widely used algorithm [110] with Python Mapper as a very successful Python implementation [82]. In particular, it utilises filter functions to assess and analyse dataset by applying dimensionality reduction [27], where data points are linked via specifically identified clusters, which provides a topological data summary [29].

Since the Dionysus and Mapper algorithms are based on the point cloud properties of datasets, the data is to be embedded onto a specific coordinate system. Furthermore, Mapper allows the analysis of two-dimensional and one-dimensional datasets, which enables a more efficient method for data analysis. However, the Dionysus library has some limitations in the construction of an alpha shape filtration [48], which makes Mapper algorithm and Python Mapper solutions more suitable compared to the Dionysius library.

The Manifold Learning algorithms contained within the Scikit-Learn package require the embedding of the data onto a low dimensional sub-manifold, as opposed to the Mapper algorithm. Furthermore, the corresponding dataset must be locally uniform and smooth. However, the Mapper algorithm output is not intended to faithfully reconstruct the data or reform the data to suit a data model, as it provides a representation of the data structure. The Manifold learning and Mapper solutions provide a useful set of data analysis tools, which enable a suitable representation of data structures and they can be selected once the structure of the corresponding data set has been identified. Since both libraries are native to the Python programming language, this allows an integration with other popular data science Python packages. The Mapper algorithm has been extensively utilised for commercial data applications, due to its capability of analysing large datasets containing over 500,000 features. This also allows the analysis of Big Data without deploying Hadoop, map-reduce and SQL database, which provides further flexibility and reliability.

2.8.9 Python for Topological Data Analysis

The extraction of meaningful and actionable information from Big Data is at the core of much of the research in this field [128]. Organisations and companies are likely to store enormous quantities of data, which are continuously produced by their external and internal processes. Understanding and assessing the value and relevance of the actionable information contained within this wealth of data is a key challenge.

Topological Data Analysis (TDA) is a new research area, which utilises topological concepts to classify and analyse data [27]. Broadly speaking, the focus of TDA is on the structure and *shape* of data, in terms of the interconnections between its components. One aspect of topology is the ability to classify objects based on their common properties, or in other words, those exhibiting invariant features. An example in topology is that by stretching and deforming a doughnut, it can be turned into a coffee mug, as depicted in Figure 2.8.1. There is a "hole" in both the handle of the mug and at the centre of the doughnut.

In many clustering and classification algorithms, the geometrical properties of the data investigated play an important role. Many such approaches focus on the concept of distance, which has to be defined according to the mathematical spaces where the data is embedded [18]. On the other hand, TDA focuses on the notion of shape and connectivity that such data exhibit. As in the example above, although a doughnut is different from a coffee mug, they share the common property of having a single hole in them, which can be used to classify them in terms of their connected components [18]. No matter how much they are stretched, they will still be characterised by the existence of such a hole.

Topology is usually specified by the metric of its corresponding space. Therefore, there is no strict requirement to limit to a specific coordinate system, thus allowing a more flexible approach [18].

A crucial aspect of TDA is the concept of *persistent homology*, which focuses on the identification of the topological properties which remain invariant [40]. Figure 2.8.2 depicts three separate datasets, which have in common the distinctive (topo-

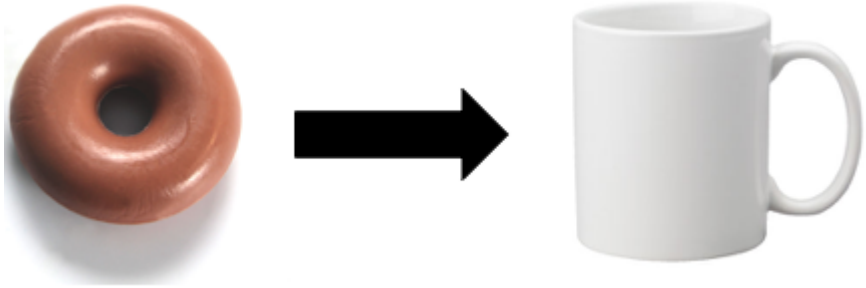

Figure 2.8.1: Transforming a doughnut into a coffee mug

logical) hole. Despite being different at a micro level, they could be regarded to be sufficiently similar. Persistent topology allows a classification of datasets based on such properties [131]. In particular, it is characterised by homology groups, so that two shapes are considered similar if they are isomorphic equivalent. In other words, datasets which can be topologically deformed and stretched into the same shape would fall into the same category. The three datasets in Figure 2.8.2 could be intuitively interpreted as referring to a similar scenario, with some added noise. TDA allows noisy data to be addressed in a more efficient manner [27]. On the other hand, the two objects depicted in Figure 2.8.3 are not topologically equivalent, as they have different connected components in terms of their number of holes.

One of the fundamental concepts of persistent topology and homology, in general, is simplicial complexes. These are space triangulations defined by combined, non-overlapping polyhedra, covering a topological space. A trivial, yet informative example of a space triangulation is image pixelation, where a real image is covered with pixels, to provide an accurate representation. One of the most important aspects of simplicial complexes is that fact that they provide an "approximation" of the object they are covering. Examples of triangulations include Voronoi diagrams, Delaunay triangulations, Vietoris and Čech complexes. Broadly speaking, they are defined by either a specific distance or in terms of ball intersections whose centres

are the data points. For more details, refer to [40]. Furthermore, the adjacency graphs generated by these triangulations can provide a variety of information on their invariant topological properties, and therefore relevant to TDA investigation [27].

2.9 TDA Methods

The aim of the following is to provide a survey of the Python implementations which can be used in TDA. This chapter follows the following structure: in Section 2.9 the existing technology is discussed, and Section 2.10 will discuss specific features of some Python methods and algorithms currently used in this research field. Section 2.11 will present the final remarks and future research.

In [32], the authors discuss the application of the mapper TDA to noisy computer network data collected by a "darknet", intending to identify activities such as port scanning and DDoS attacks. In particular, they consider a dataset comprising over 3 million data points, which would prove difficult to visualise using traditional methods. Previously, software packages such as Suricata [115] would be utilised. However, it would not be sufficiently robust to successfully address noise whilst identifying all the attack patterns. The default Mapper code has been recently extended with KeplerMapper [4] for Python 3 (instead of Scikit-Learn [1]), and utilises C for the DBScan clustering to improve efficiency.

Motivated by the fact that large unstructured data sets can be visualised to provide valuable information, in [102] a commercial implementation based on TDA is described. Its objective is to provide a reliable diagnosis system to analyse large biomedical datasets, which has been shown to have high accuracy. To achieve this, TDA is used to define specific artificial neural networks coupled with Kolmogorov-Smirnov test [125]. The software package Ayasdi [11] is another commercial package to provide topological data analysis.

Biomedical data are often incomplete. Many analysis methods suffer when incomplete rows must be omitted, as valuable data might be subsequently removed. In comparison, the use of TDA does not require incomplete rows to be removed

and can, therefore, produce information using all available information.

The use of TDA as the first stage in data analysis enables a wider range of initial data to be sampled, creating a more detailed grouping of data, regardless of any missing or incomplete fields. TDA allows for generality, as any notion of similarity can be successfully exploited in this context. On the other hand, classic machine learning algorithms typically require a comparably high level of similarity to produce any meaningful output.

2.9.1 Topological Algorithms

The methods discussed above fall into three main categories: persistent homology and Mapper and Manifold Learning, which will be discussed in this section, with a particular focus on their Python implementation.

As described in Section 2.5.3, persistent homology is an algebraic method to assess topological features of shapes, via suitable triangulations, a simplicial complex is defined in terms of a distance function of the underlying space [49]. This algorithm is relatively robust and well tested in the academic field [39]. An implementation of the persistent homology algorithm is found in the Python library Dionysus [80], which contains a variety of algorithms (Lower-Star, Vietoris-Rips, etc.) to cater for a large number of data shapes and types. Furthermore, Dionysus is a Python interpreter, which has been shown to be computationally efficient (but slower than a pure C++ implementation), whilst producing accurate data representation [80].

Manifold Learning, allows data of high dimensionality to be visualised by extracting key underlying parameters which form a low dimensional manifold. Manifold Learning has various approaches, each implementing the extraction of data in a slightly different manner. The ability to utilise many of these methods can be achieved through the widely used and well document Python library Scikit-Learn [1]. This Python library allows Locally Linear Embedding, Modified Locally Linear Embedding, Hessian Eigen mapping, Spectral Embedding, Local Tangent Space alignment, Multi-dimensional Scaling and $t-$distributed Stochastic

Neighbour Embedding. With such a wide and carefully implemented set of Manifold Learning algorithms, it is possible to carry out TDA allowing for rapid and accurate data analysis of almost any data type.

As discussed earlier Mapper is a widely utilised algorithm [110], which has also been successfully implemented in Python as Python Mapper [82]. The Mapper algorithm uses multiple filter functions to allow a coordinate system to represent a set of data, thus reducing the complexity of the dataset via dimensionality reduction [27]. Data points are connected via non-empty intersections between identified clusters, resulting in a topological summary of the data [29]. The Mapper algorithm can be successfully applied to cloud point data analysis, and it offers an alternative solution to the Dionysus implementation. However, the benefit of Mapper compared with Dionysus is that its implementation is in pure Python as opposed to a port from C++. This creates a more efficient and more manageable code when used in a live environment.

2.10 Algorithm Efficiency and Performance

As discussed above, the Dionysus and Mapper algorithms focus on point cloud datasets, which implies that the data must be embedded onto a specific coordinate system. Mapper also supports two-dimensional and one-dimensional data sets defined as two-dimensional vector input data and one dimensional pairwise distances, respectively. This approach enables a more flexible and efficient method of analysing data.

Even though the Dionysus library also allows two-dimensional inputs, it introduces a limit in the construction of an alpha shape filtration [48]. As a consequence, from a data analysis point of view, the ease and flexibility of the Mapper algorithm and Python Mapper solution is preferable to the Dionysius library.

The Manifold Learning algorithms contained within the Scikit-Learn package require the data to be embedded into a low dimensional sub-manifold, as opposed to the Mapper algorithm, which allows higher dimensionality. In particular, they need the dataset to be locally uniform and smooth often with a restriction on sam-

pling uniformity. In contrast, the Mapper algorithm output is not intended to faithfully reconstruct the data or reform the data to suit a data model, as it provides a representation of the data structure.

The Manifold learning and Mapper solutions provide a useful set of data analysis tools, which enable a suitable representation of data structures. Furthermore, they provide different methods based on specific assumptions on data inputs and output. Therefore, a choice between them can only be made once the corresponding data set has been created and its structure is known. The ease of implementation due to both libraries being native to the Python programming language allows an integration with other popular data science Python packages. This facilitates the creation of an efficient and computationally feasible data analysis platform.

The Mapper algorithm has been extensively used for commercial data analysis tools, due to its ability to process complex data sets containing over 500,000 features. This also enables complex Big Data to be handled without the requirements and complexity of deploying Hadoop, map-reduce and SQL database, prove the flexibility and reliability of the algorithm. Figure 2.10.1 shows details of the performance of the above algorithms.

2.11 Manifold and Mapper Method considerations

The manifold learning and Mapper methods for TDA allow greater flexibility and data input sources compared to the Dionysus implementation. Furthermore, highly specialised data scientists have been contributing to the Scikit-Learn and Mapper projects, creating a peer-reviewed and well-maintained implementation, which is not the case for Dionysus. Industry Standard analysis tools, such as Ayasdi, have been built with the Mapper algorithm at their core, demonstrating their industry appeal and credibility. This suggests that data analysis platforms aiming to obtain efficiently interpreted results with applications to a decision-making process should prioritise the implementation of the Mapper algorithm.

2.12 Summary

Data has become increasingly crucial within the majority of data-driven fields, and as a result, has attracted considerable attention from multi-disciplinary research areas. The need to provide cutting-edge algorithm and methods to address the multiple challenges posed by the diverse and large quantity of data continuously created has led to innovative solutions and approaches. This chapter focused on a survey of specific research areas, including Network Theory, Bayesian Networks, NLP and Machine Learning, which have enhanced our capability of identifying, extracting and assessing actionable insights from Data.

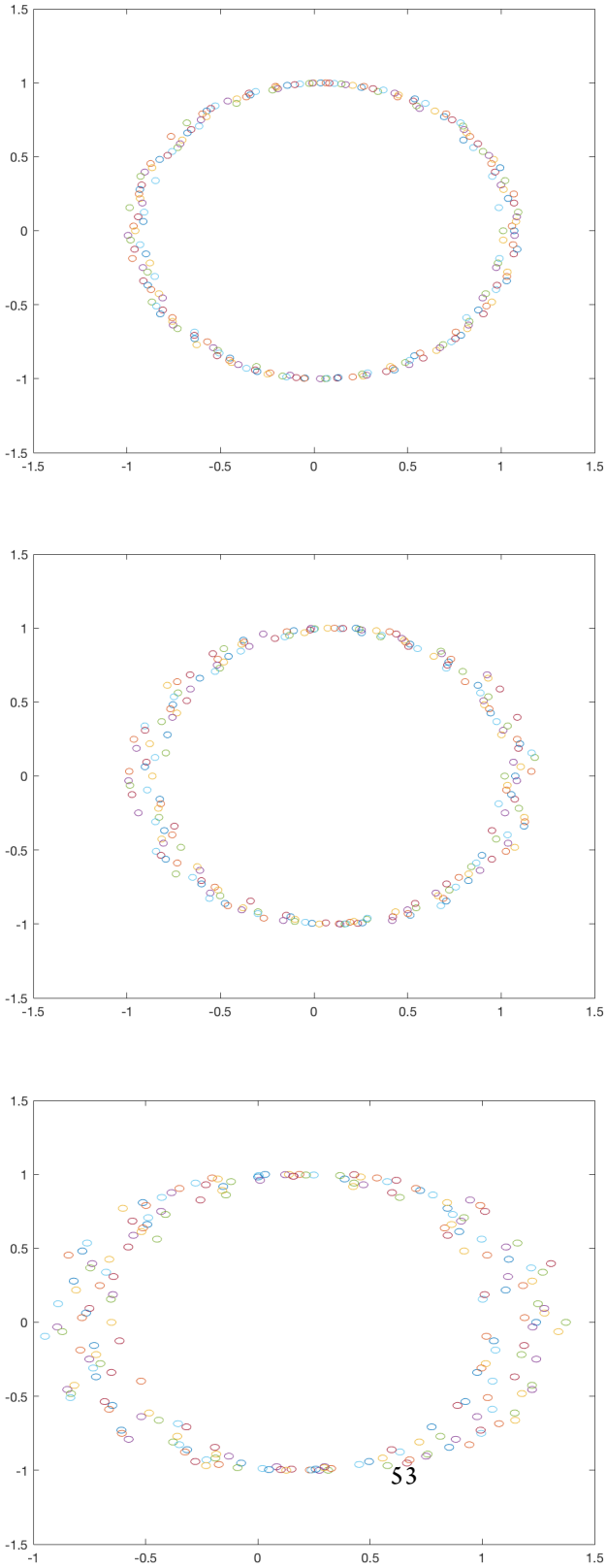

Figure 2.8.2: Three datasets with similar topological properties. [95]

Figure 2.8.3: Two sets with different topological properties. [95]

Method	Properties	Limitations	Strengths
Python Mapper	Native Python Combines Filter functions, the Mapper algorithm and visualisation results	A pure python implementation, slow than a traditional programming language implementation such as C	Robust algorithm capable of relative equal output regardless of sample size. Easy integration with Other Native python libraries
Dionysus (Persistent Homology)	C++, with Python bindings Persistent homology computation, Vineyards Persistent cohomology computation, Zigzag persistent homology.	Difficult installation, requires a large list of dependencies Not all functionality is completely available in Python	Highly resistant to data noise.
SciKit Learn (Manifold Learning)	Native Python. Non-linear dimensionality reduction. Wide variety of possible approaches.	Same scale must be used over all features - Manifold learning uses nearest-neighbour search. Certain input configurations can lead to singular weight matrices (more than two points identical)	Able to adapt Linear Frameworks to be sensitive to non-linear structures in data. Comprehensive community support Integrates seamlessly with widely used Machine learning libraries already available in Python

Figure 2.10.1: Performance data for networkX and Graph tools. [95]

3

Social feed Adverse event discovery

In this chapter, the term Adverse events refer to an adverse social media event not that of a clinical adverse event.

This chapter discusses how adverse social events can be classified via topic modelling and mitigated through management policies, the process of analysing social media text feeds has the abilities to discover these events, this chapter creates an initial solution on how to classify an adverse event and how to manage the events once discovered. Without managing what is classed as an adverse social event it would not be possible to produce machine learning that is capable of discovering the adverse events. The guidance given can help business and organisations to discover what social media topics and events would require this level of response and adjust their definition of an adverse event. The chapter contains discussions on event identification from social media feeds via topic modelling and the use of

dropping common terms to aid topic modelling. The final part of this chapter looks at patient emotive and how this influences the data found during social media text extraction.

An important aspect of managing hospital resources is the ability to predict and supply a suitable amount of staff to reducing waiting times and provide a quality service to all patients and families who interact with the hospital during this time. Research into event identification in relation to hospitals and patients is focused around drug interactions and interactions within the hospital, to successfully build a machine learning approach towards identification of social events the currently used methods must be understood and adapted.

3.1 Event identification from social feeds

By utilising sentiment analysis and network theory, emerging and current trends can be identified, allowing for action to be taken before the event escalates into a large crisis or develops into an event. As new topics are discovered, topic sentiment can be used to help judge how important the discovered topic is. Whilst sentiment cannot be used as a sole identifier of the importance of a topic, sentiment can help build stronger analytics when combined with other topic identification methods [134], [103] have methods for real-time identification of topics. The Topic Sketch solution provides the flexibility and speed to monitor real-time social feeds and provide immediate and reliable topic identification, the approach taken by the authors to create data sets based upon every 2 and 3-word pairing or triplet, this approach whilst fast does require a significant memory base to successfully computer the elements in a near real-time manner (> 16GB).

In the first instance, it is preferable to utilise an existing system for topic and event discovery, however, should none of the existing methods investigated proves suitable, an alternative option is to create and implement a system based upon groups of 10 tweets each group iterated over via named entity recognition and with custom code designed to gather related terms via naive parts of speech process.

This process could be repeated storing sets of tweets into groups and comparing each group with the building topic cloud. The similarities between sets of 10 will increase the importance of a topic allowing for new topics to not be buried but for recurring topics to show importance. Topics can be grouped or split by sentiment, creating granularity over a single topic that may contain opposing viewpoints, adverse events are likely to contain strong sentiment identification of sentiment spikes, rapid changes, or sustained strong polarity sentiment can be used to triggering event management protocols.

To apply the system to a health care environment it requires a consideration of medical terms and their ordinary English counterpart, along with geolocation fencing to ensure only relevant local topics are monitored. It is important to discover traits and common features that conform with an adverse event to help train and refine the models to ensure that specific targeted events related to emerging health care issues or events that could impact the hospital can be identified. By defining parameters of what will be classed as an adverse event it is possible to manage a media stream and monitor for unusual events. Events can then be managed, and action is taken to remain in control of an event or allow the organisation to address an issue that has been identified. The ability to quickly identify and pro-actively address adverse events is a highly sought after to ensure negative events and press, the use of manual review could conceivably be replaced by machine learning to not only reduce the human burden but to cover a wider area of social feeds and data sources that could not be otherwise be monitored without significant staffing costs.

3.1.1 Adverse Event identification

Murff et al. in 2003 [81] discuss how the 'gold' standard for identifying adverse events in many patient studies has been via manual chart review and voluntary reporting of incidents. This method can be accurate but is highly dependent upon the reviewer with a time burden placed upon highly skilled reviewers who would make better use of such time on more important issues. A problem with data con-

sistency between in and outpatients also exists with outpatient charts being significantly less data-intensive than inpatient charts making them more challenging to discover adverse events via a chart review. The report details manual methods, combined modalities, automated detection systems and cognitive frameworks, concluding that the most prominent method and 'gold standard remains the manual review process. An area of importance in-hospital adverse event recognition and prevention centres around pharmaceuticals, [17] discuss how adverse events from pharmacy and clinical lab data (two common sources of coded data), supply direct data that can be managed by machine learning / natural language processing. Implementing such a system to reduce and eliminate dosing errors and clinical values that are out of range. Whilst these two errors are not found within the social feeds that this project aims to address, the problems and solutions appear to be common between the two elements, an area of growth for the project that would be welcomed by those within the clinical and hospital setting.

3.1.2 Identification Methods

Utilising an implementation of named entity recognition or a neural network trained on standard non-medical data creates a system that does function, however, [64] demonstrate the importance of supervised learning utilising carefully selected medical databases to ensure a quality training dataset. In [133] clinical named entity recognition is tested utilising a specialised corpus *i2b2 2010 clinical concept extraction corpus*, the system built uses a Recurrent Neural Network (RNN) and outperforms manually defined and unsupervised methods with a strict F1 score of 85.94%, however, the difference between all methods F1 scores ranges from 77.33% to 85.94% an 8.61% increase in performance. This increase demonstrates that without hyperparameter training the RNN shows improvement, the authors also note that no performance enhancements have been utilised for the testing and that the Java implementation of each algorithm was used along with the Nvidia Tesla K40 GPU. The computing system used to process that data using the RNN significantly outperforms most systems available for daily use in a hospital set-

ting, such a system will be unobtainable in the clinical environment, as such more research into performance on a low powered system will be required to understand how system performance will impact F1 score and near real-time processing. A range of methods already exist and could be used to help reduce computational burden and improve overall computation speeds on a high powered system along with low or underpowered computers, these methods including stemming both prefixes and suffixes to improve the lexical match, mapping to a thesaurus such as the Unified Medical Language System (UMLS) Meta thesaurus or MEDLINE to associate synonyms and concepts, and simple syntactic approaches to handle negation. Whilst not scoring as high as RNN Natural language processing [112],[45] promises improved performance when compared to an expert human encoder by better characterizing the information in clinical reports. Two independent groups have demonstrated that natural language processing can be as accurate as expert human coders for coding radiographic reports as well as more accurate than simple keyword methods. Whilst a computerised approach has many benefits it is not without its unique limitations, the data must be presented in a digitised format, any information that is to be processed my first be converted into a standard encoded text format, the process of this could take many forms, such as additional machine learning for character recognition or manual input to a system, without the conversion to a text format the system will not function, unlike the manual human counterpart who can perform equally on handwritten as computer written data. The increase in processing power will come at an increase in cost, the manual alternative whilst slower does not come with a scaling cost to perform the task. Significant research has been conducted related to NLP / ML for adverse event discovery using a drug interactions database to cross-reference adverse effects. It is noted that no proof of concept code or working implementations have been released for academic scrutiny or released in an open-source manner. Drug to patient cross-reference systems have been shown but open adaptable solutions are not widely available as such a system that capable of adverse drug interactions would be of research interest. Can prevent drug misuse but must be fully automated to be effective. A method discussed by [138] involves training a classifier on

a small set of labelled posts/documents that match each category. A possible solution is to recreate a similar classifier with a small set of training data related to each topic, the topics and data would need to be established with the support of specialists, then by using a neural network and a range of historical data should allow for a fast prototype that gives reasonable accuracy allowing as a workable prototype. An option for event identification is the selection of domain-related bigrams, unigrams or n-grams each domain will require its own crafted set, with the most suitable not known until testing and evaluation has taken place. Whilst terms that are common across all domains but add no value for example ('pray for', 'help') must be removed see 3.2. A debate about the overall performance of supervised vs unsupervised text classification is still ongoing, with the most common answer is a domain and task dependant answer, certain domains are more easily solved with one or the other, results from [65] suggest that with supervised domain-specific training will outperform an unsupervised system. The initial prototype will focus on an input that will vary in complexity so a more general prototype will be developed, the system must avoid generating a complex model that cannot generalise based upon a low feature count <100 labels as in this case, a manual selection of the most informative words to be used as features can be unreliable. A solution for low label data is the use of word clustering (t-sne) and to then use those clusters as features. This could allow words that do not appear readily or are rare to be successfully grouped allowing for words that have not even appeared in the training set to be successfully grouped. With greater training data and available data points the ability to train a model accurate and capable system becomes available, this trained and development will only become available after an initial prototype with the greater data, however, it is proven that over 1000 labels for Bag of words [20] will outperform smaller datasets. Below that limit, the cluster method is better.

3.2 Dropping Common Terms

The dropping of common terms is identical to the removal of stop words, common terms that add no value to the hospitals' goal of discovering topics that will affect admissions to A&E. In order to discover unhelpful terms, manual review of the data and selected top terms that are unhelpful to the final analysis are removed. Re-run analysis and repeat process to ensure clear visualisations. Research [105] suggests that the removal of stop words or common terms should be performed after the model training and before the presentation of the data. The removal of stop words is more beneficial for the result of data display than it is to the training of the model little to no change takes place when these stop words are removed before or after training [105]. The highly specified nature of the stop words should not cause an issue, it will, however, limit the cross-use of the system. Requiring additional stop words generation for other domains.

3.3 Suitable real-time text analysis via clustering

[107] illustrates the real-time analysis of large scale data from a twitter feed using sentiWordNet, polarity wordNET, with back end support from apache spark and apache Hadoop creating a distributed computation platform (programmed in Java). They demonstrate the concept of real-time processing on a social feed, a similar process is followed in this work with more emphasis on the final analytics and low powered computing rather than the distributed computing via Hadoop to creating a usable meaningful and actionable outcome for the hospital. The implementation used by [107] benefits from the distributed computing Hadoop and spark, its only analytical task is sentiment analysis, the ability to run across multiple systems via Hadoop and spark may be important factors in the ability to run real-time analytics on streaming social feeds.

3.4 Suitable management policies to address the impact of social adverse events

Across high profile business and organisations, the ability to quickly react and adapt to adverse events. Within the NHS these events have various levels, from PR events to medical events, each much be managed and carefully assessed, the business impact of each event could be critical in for public image. Events could also result in a financial loss of significant proportion. None of the outcomes from an adverse event results in positives for the organisation. To ensure that minimal impact results from the event it is important to react appropriately and promptly. Having a range of policies and procedures in place that can be enacted immediately upon discovery of an incipient adverse event or in the early stages of an event, should support the careful management and de-escalation of such event. To ensure an adverse event does not become critical.

A review from [126] highlights management policies. The conclusion provides key areas of note, these have been adapted to provide guidance and management policy considerations in the implementation of social adverse event resolution and response.

1. Social media engagement should be integrated into risk and crisis management policies and approaches. Renewal of the crisis communication plan should have a section for communicating with stakeholders and working with the media. Social media can be used to both communicate directly with stakeholders, the media and the public at the same time. More importantly, social media provides a built-in channel for stakeholders, the media and the general public to communicate directly with the organization. Incorporating social media into the plan ensures the tools will be analysed and tested before the crisis and requires continual updating of the communication plan as social media evolves.

2. Incorporate social media tools into routine risk assessments. Social media allows the organisation, to listen to the concerns of patients and other risk

bearers. When users create and manage their content, external and internal social media monitoring (listening) is crucial. Also, tracking issues through social media and providing the crisis management team with the reports can increase the potential that a crisis will be addressed sooner and demonstrate to the team why social media needs to be embraced in the crisis response.

3. Engage social media in daily communication activities. Individuals may have information that is crucial to the mitigation of the crisis, but if they do not trust the organization or even know where to find it, that information will likely not be shared. To build partnerships and build trust, the discussion with the public should already be taking place. Do not begin a social media campaign or attempt to begin social media interactions during a crisis. Internal social applications, such as live chat, blogs, or wikis can streamline inter-organizational communication and increase efficiency, begin implementing alternative communication channels early to enable users to be familiar with the system and to build inter-team relationships that could prove critical during an adverse event. Involving the crisis management team in the development of the crisis plan and document management site through social media, rather than handing the task off to a single individual, increases the potential for interactivity in the crisis response.

4. Join the conversation, including rumour management, and determine best channels to reach the segmented public. Having a profile on a social media site is not enough, interaction is key. While an employee can track issues, interaction is essential in addressing misinformation and establishing the organization as a credible source. Responding to posts demonstrates the organization cares what patients think and can be trusted to address their concerns. Reaching specific groups with a key message is a foundation of targeted communication. In crisis communication, practitioners often resort to the standard mass media push to reach everyone at once. Crisis communication must still consider how messages will be interpreted and its appropriateness to the target group, whilst understanding who will not be

reached. Determining the best communication channels for target groups such as offline, online, or in the community should be incorporated in crisis communication.

5. Check all information for accuracy and respond honestly to questions. Inaccurate information shared and retweeted makes the hospital look unprepared and disorganised. While it is easier to simply skip over a post to which you do not want to respond or to ignore press requests for comments, the public, like the media, will turn to other sources if the hospital refuses to comment on key issues. If you do not know the answer, it is better to communicate the uncertainty of the situation and explain what you are doing to find out the answer than to answer incorrectly or to not answer at all.

6. Follow and share messages with credible sources. Collaborating with trustworthy and supportive sources can not only embellish the credibility of the organization but also, increase the reach of the organization. By cross-posting and retweeting, messages among partner organizations, a coalition of credible sources are established, and more individuals are reached through the shared networks. This gives social users a network of trust for reliable information relating to the hospital or its services.

7. Traditional media is already using social media. The crisis will likely be discussed through social media, and traditional media will be part of that discussion. If the organization is not engaged and active in the conversation, the media will find other sources through social media to comment on the crisis. Thus, when it comes to being accessible to the media, not engaging in social media can have the same effect as not returning a reporter's call.

8. Social media is interpersonal communication. Social media allows for human interaction and emotional support and is important to patients dealing with crises. Generic marketing blurbs will be seen as cold, callous, and impersonal and will not encourage the relationship building and mending needed in a crisis. The hospital should be ready to incorporate and respond

to emotional appeals that can and do appear rapidly on social media, overreaction should be avoided but it is important to quickly acknowledge a situation and take reasonable steps to demonstrate organisational awareness of a sensitive issue.

9. Use social media as a primary tool for updates. Organizations often promise to follow-up with the media and public as soon as they have new information, and yet, they wait to release that information until a press release can be drafted, refined, and sent out or posted passively to the organization's web site. Or, to convey the emotional concern required, wait until the next scheduled press conference. These responses can and should still take place, however, using social media for updates on the crisis response and recovery allows the organization to humanize the response and continue to be a reliable source, without requiring all the exact details and time needed to fill a press release or hold another press conference.

10. Ask for help and provide direction. Giving people something meaningful to do in response to crisis helps them make sense of the situation. As a partner in the crisis response, the public can provide essential information. By providing that information, social media users are acting. When an organization requests useful information via social media, it helps both the hospital and the social users who respond in managing the crisis. If there are actions individuals can take to reduce risks or assist in the recovery efforts, social media is an ideal forum for reaching target groups with the directions needed. Even more, by simply forwarding, cross-posting, or retweeting the directions, the users are acting.

11. Social media is still just another media channel despite its technological advancements, rapid access to information, large numbers of potential views, low cost, and ease of use. The power to communicate remains with the hospital team, their behaviours and narrative content, not in the technology. The real value of any communication – social media included – remains

the quality of the content being disseminated around the actions a brand or company is taking, the empathy for affected stakeholders being displayed, and the appropriateness and relevance of the context and perspective are provided. Social media is a tool that can assist practitioners in following the best practices in risk and crisis communication but is not and should not be a total solution.

3.5 Recomendations

The importance of a timely and meaningful response to adverse events does not focus on one element, the hospital and associated teams must manage a wide range of media assets to correctly convey the right message whilst ensuring a suitable time frame for response. With too slow a response a distant, cold and uncaring hospital will be perceived, but a rushed response with inaccuracies convey a hospital that is unorganised and incapable of communication, some recommendations are discussed below.

- Start with owned properties, social media presence, website, applications (iOS/Android). These allow users to receive information as it becomes available, keeping users informed about a situation or responding in the first instance help demonstrate awareness and maintain positive relations.

- All account guardians must maintain professional but personal communications ensuring a consistent presence regardless of the platform, it is expected that real-time social platforms will have a less formal approach but will still be communicated professionally.

- Define a clear scope & operational definitions – clear expectations as to what constitutes an adverse event, some interactions may be hostile but not constitute a full adverse event, size and scale must be defined for when crisis / adverse event plans are brought into action.

- Trained staff – Ensure staff are familiar with the operation of a social media account, easy mistakes can be made by sharing information that was intended to be private. Training on social media management platforms will allow staff to monitor and contribute to multiple accounts across social platforms from a consistent interface.

- Make use of quality assurance methods already in place, the hospital already ensures quality across interactions with patience's existing protocols and policies can be adapted to ensure quality social media interactions.

- Staff who monitor Social platforms must be prepared for large volumes of small scale event as social platforms are easily accessible by anyone, limiting the interaction time for a social media manager can help prevent burn out responding to a large scale influx of questions and comments.

- Use available insights – the social graph of users who interact with the channel can help with carefully tailoring the language and content posted to ensure the greatest reach and interaction, creating positive experiences can often help with reputation.

- Response – When an adverse event is identified, a quick and decisive response is important, even if this response is just to gather information about the event in a more private setting. Creating a set response time can ensure users feel valued. Always follow the platforms guidelines and best practices ensure a human voice of the brand, with a personalised greeting, if trained staff have strengths, they can handle certain types of adverse events. Ensure staff who have some experience or knowledge of the event handle the adverse event.

- Once you are clear on the basic message, you need to decide how to deliver it. That means creating guidelines so that anyone writing a social media post knows what's expected of them.

- Determine rules for communicating with key stakeholders and executives. It is important to understand the chain of command and authorisation channels who need to be involved in an adverse event situation. A clear structure for who is responsible for activating a plan must be available to social media account guardians, it allows for the crucial rapid response.

- Set network-specific guidelines for communicating on social media, certain platforms rely on a media type (such as images or video), it will be necessary to prepare images and that can be used to convey a particular event such as closures or announcements with official branding, the pre-created images.

- Decide on a process for communicating updates via your website and other online company channels not covered by social media.

- Create guidelines for employees outside of the crisis communications team advising how to respond to inquiries. Ensure only essential persons are delivering the information from authorised accounts, staff should be aware that posting information can be counterproductive to the situation.

- It is ok to suspend non-essential social feeds and to restrict timed events and scheduled posts, a clear calendar of events must be maintained to ensure that all social managers are aware of what may be posted due to a scheduled event as sometimes it is important to avoid response.

3.6 Patient Emotive

Visits to a hospital or clinical environment are often emotional, this emotion often comes from the close family members of the patient especially those who are in a vulnerable position such as children or elderly. The emotion felt by parents taking children into hospital can have an impact on the decision-making process and the actions that might be undertaken by the group that would not normally be acceptable. The ability to discover adverse events that have been created by an emotional response will require a more empathetic and controlled response than

an error with parking charges or lack of refreshments at the canteen. Understanding patient emotive and its identification can help diffuse situations. Patients and families are going to be more emotional and have a heightened emotional response during treatments, with more severe treatments and medical issues causing a more significant emotional response. Most research around patient emotive is focused on clinical error and patient to caregiver interactions, the use of machine learning could help discover if a similar level of emotive is visible during social media interactions with the hospital. Recognition of the emotional impact of patient safety incidents (and medical errors in particular) on both patients and healthcare professionals is growing [111, 132]. The physical, emotional and financial trauma experienced by patients and the powerful emotional impact of the error on healthcare professionals has been described by many authors. [63, 108, 111] Health professionals report significant emotional distress in the aftermath of making an error, and in particular, feelings of shame, guilt, fear, panic, shock and humiliation [53]. This distress readily transfers into personal life, creating additional burdens, such as inter-personal conflicts and sleep disturbance. In the workplace feelings of distrust, reduced goodwill and detachment from patients are all described [106]. The ability to detect adverse events can enable hospitals to provide care and assistance to healthcare professionals, automated detection of potential problems can create preventative cover rather than reactive cover.

3.6.1 Emotion as a Contributor to Patient Safety

During a recent [38] high profile adverse event at Alder hey hospital, it becomes clear that not only did patient emotive cause distress it also‘ escalated [109] into unacceptable risk for staff and other patients. The emotion of this incident had the potential to influence and degrade services to other patients and cause stress and discomfort to staff. The protests caused by social media had a lasting impact on the hospital and the dedicated staff who had to endure untold abuse via social media and in-person for performing their duties. The ability to monitor collective emotive can help hospitals manage risk towards staff and patients. The effect of

patient and social emotive on clinical outcomes and staff wellbeing has not been extensively researched the use of machine learning could aid in generating data for future works.

3.7 Summary

This chapter has discussed the process of adverse event discovery in social media within the health care setting and the hospital environment. The importance adverse event management is to organisational reputation and public appearance, for machine learning to be of use in the identification of adverse events some suggested policy changes are meaningful amendments proposed that could be implemented alongside existing policies to help organisations prepare for and minimise social adverse events, whilst these changes are not specifically related to the use of machine learning and more aimed towards social media usage it is important that these changes are considered as the ability to search a large array of social feeds and analyse them in near real-time creates opportunities on social platforms that would not normally be discovered. The chapter details machine learning methods that could be implemented for adverse event identification, with some experimental results demonstrating that machine learning algorithms exist and with some tuning of hyperparameters a system could be implemented that will run near-real-time analysis on social feeds providing detailed analytics on a low powered system, a low powered system is on that is not boosted by GPU processing as is the industry standard for machine learning systems. further discussion on the methods used is expanded upon in the first validation.

4

Automated Approach to Hospital Data Analysis

This chapter develops a preliminary approach to determine and assess the contributory factors in patient satisfaction, based on an automated knowledge extraction method from articles from PubMed and social feeds such as Twitter. This chapter includes the proposed method, the dataset used and an evaluation of the system. The chapter is structured as follows: in Section 4.2, the proposed method is introduced, and Section 4.3 describes the dataset used in this work. Section 2.10 focuses on the implementation and validation results, and finally Section 2.11 concludes the chapter.

Hospital data refers to the gathered information from PubMed and social media feedback relating to the hospital collected from Twitter.

4.1 Hospital Data

Hospital data size is likely to increase dramatically in the coming years [31]. It is therefore vital for healthcare establishments to use the existing devices, structure and methods to control big data efficiently to potentially avoid large revenue losses [66]. Previous and current research on data analytical methods had been found to be effective on a huge amount of un-analysed health and medical patient data. During discussions with members of the Innovation Hub at the Alder Hey Children's Hospital, it was confirmed that the NHS currently identifies most of its feedback based on a patient experience survey along with the friends and family test. However, current literature suggests that the use of the friends and family test is limited due to the small range of patients that willingly complete the form, who are usually negatively biased [88]. Other methods include patient panels, focus groups and the mystery shopper data collection style often used within retail. All of these methods require lengthy processing time [3], which increases the action time between collation of feedback and action being taken. Any delay between feedback and action in a hospital setting can result in patients feeling being ignored [101]. From a review of published sources, it is apparent that much of the available literature is of a non-experimental, descriptive nature. Such a trend is not uncommon for research questions that address the effectiveness of organisational processes, and the information gathering questions greatly vary if they are posed to managers, clinicians, or patients. This issue raised by many reports, creates a divide between perceived good health care from a patient's point of view, poor service provided by a clinician and vice-versa [74]. Only a small amount of literature exists that directly addresses the action feedback loop in a hospital context, i.e. the feedback process by which incident data in any form are transformed into beneficial improvements in operational safety or actionable events that produce positive improvements in patients' experiences. An academic review of patient satisfaction surveys reveals that measuring patient satisfaction has numerous problems, as most surveys oversimplify complex issues that each patient faces during his or her hospital stay. Furthermore, the notion of *satisfaction* is highly debated,

which is defined as a judgement formed by individuals over time as they reflect on their experience. This experience can differ from quality health care as perceived by a health care professional [37]. Most surveys or feedback gathering exercises have a long processing and administration overhead, leading to lengthy delays between comments, suggestions and experience reviews from a patient transforming into visible action, usually long after the patient has left the hospital. In [25], the authors discuss a qualitative interview study concerning the effectiveness of data feedback in supporting performance improvement efforts in eight US hospitals. Data quality, timeliness and credibility were identified as important factors for effective improvement, along with leadership and persistence in data feedback processes "*data feedback must persist to sustain improved performance. Embedded in several themes was the view that the effectiveness of data feedback depends, not only on the quality and timeliness of the data but also on the organisational context in which such efforts are implemented*" [25]. In [61], the authors highlight two critical determinants for success: timely, effective feedback and demonstrable utility. The former assures reporters that their reports are acted upon and are not trapped into an administrative "black hole". Demonstrating the local usefulness of incident data, in addition to the development of external reports, influences user adoption and compliance, and can improve reporting rates. These factors highlight the need for immediate analysis and processing of patient feedback, clinical and hospital data. The ability to follow the action in a timely manner is further highlighted by [47], who discuss the importance within US health care of follow-up actions after a safety or error report. The authors suggest that more emphasis should be placed on event follow-up, prioritising opportunities and actions, assigning responsibility and accountability, whilst producing an action plan to meet the needs of the reported issue [47]. Current literature treats feedback from patients as a separate category from that produced by clinical staff [67], with differing procedures and protocols in place for dealing with feedback. NHS staff training literature contains no guidance for feedback. Each hospital manages and addresses its feedback, comments and complaints uniquely, with some publishing reports after an unspecified period. In some cases, this can take upwards of four months. Reports also indicate

that formal language and taxonomy varies between the hospital setting and background of each patient who provides feedback. A standardised taxonomy to focus on specific areas is recommended to narrow feedback into actionable groups, creating manageable action groups with patient satisfaction metrics applied to each area [54]. The literature relating to hospital data collection indicates that current methods are largely ineffective at providing immediate feedback to patients, allowing the patient to feel involved and that any reports they submit will have an impact upon the conditions experienced throughout the hospital. With real-time data collection and feedback opportunities, the use of free form text entries, as well as emotional and personal wellbeing index scores, would ensure the data being collected will directly relate to the current experience. In this chapter, we discuss an initial implementation of a method to assess patient satisfaction based on text mining techniques, to populate suitable decision networks. In particular, we consider the method introduced in [116, 120], where fragments of Bayesian Networks (BNs) are automatically extracted from textual sources. Furthermore, sentiment analysis of tweets can facilitate the assessment of the influencing factors that affect patient satisfaction.

4.2 Description of the Method

For this thesis, grammar-based information extraction is utilised, which is defined by a set of *text patterns* aiming to identify textual fragments with a specific syntactic structure [75]. This determines the different parts of sentences, which refer to nouns, verbs and attributes, as well as suitably defined keywords, which may indicate a specific state [34].
Sentiment analysis is carried out to assess the mood, or *polarity* embedded in textual information [72]. In particular, the output consists of the following couples

```
(nouns, pol)
```

where,

- `nouns`refers to the concepts identified in the sentences.
- `pol`refers to the sentiment polarity identified in the sentences, and it is a numerical value between -1 and 1.

The value of the sentiment polarity has been evaluated using the VADER sentiment analysis algorithm [57]. This is a rule-based model for specifically designed to carry out sentiment analysis from the social media content, whilst generalisable to various domains.

The output is then used to populate a network, whose nodes include the nouns extracted from `nouns` connected by an edge with a weight in the range $[-1, 1]$ corresponding to the sentiment polarity, were negative and positive values signify an overall negative or positive mood, respectively. In general, we might have more than an edge between pairs of nodes as these might have been extracted from different textual data.

Using the method introduced in [120], specific (semantic) networks can be defined by identifying the different relations captured by textual fragments. In particular, the following text pattern was utilised

```
(NP1, keyword, NP1),
```

where

- `NP1` and `NP2` are the noun phrases, which contain nouns corresponding to specific concepts
- `keyword` refers to a set of keywords related to semantic dependency [123].

The above text pattern allows the identification of probabilistic relationships between concepts, which can be of *dependence* or *independence* types, depending on whether a dependency or independence relation is present between the corresponding concepts, respectively. The topology of the resulting network is subsequently investigated to generate fragments of Bayesian Networks (BNs) [116].

BNs are acyclic networks which model the probabilistic relationships between variables, as well as their historical information. In particular, their nodes are associated with concepts, which are conditionally dependent if they are joined by an edge [89]. BNs have been used in a variety of modelling scenarios defined by uncertain or partially-known information and they are particularly useful in understanding the influencing factors in a decision system.

This network is then used to create visualisations using TSNE from the word-vector space found within the network.

4.3 Description of the Dataset

The initial validation is based on a textual dataset focusing on the patients' satisfaction at UK hospitals. More specifically, the main components included approximately 3000 tweets and over 370 abstracts freely available from PubMed, which were identified by the following keywords: *patients' satisfaction, hospital,* and *NHS*. The dataset was subsequently pre-processed to remove any format inconsistencies, such as extra lines, strange characters and information not relevant to the context, including authors' names and affiliations, etc.

4.4 Evaluation

The evaluation was carried out in Python with the NLTK, vaderSentiment and NetworkX libraries. NLTK and vaderSentiment allow access to more than 50 corpora and lexical resources such as WordNet [62], which integrates text processing libraries to classify and analyse textual data and assess their sentiment polarity [22, 57]. In particular, it integrates the Stanford Parser [75], which was used in the analysis in this chapter. The NetworkX library is designed to define, analyse and investigate the properties of complex networks [52]. This was used to define the networks extracted from the textual sources, and subsequently, analyse them via the numerous methods available in this library. It contains several algorithms to fully assess the topology of networks.

A network was created based on the PubMed articles, which identified dependence relations using the method introduced in [116]. The aim was to assess whether the above concepts are strongly linked as well as how they are discussed in the tweets extracted. We identified the following concepts:

- Quality of life
- Patient's satisfaction
- Treatment satisfaction
- Diagnosis
- Readmission rate
- Medication Management
- Cost-effectiveness
- Age
- Care Management
- Experience of care
- Social requirements
- Vulnerable patient
- Improvement of services
- NHS
- Staff satisfaction
- Wellbeing.

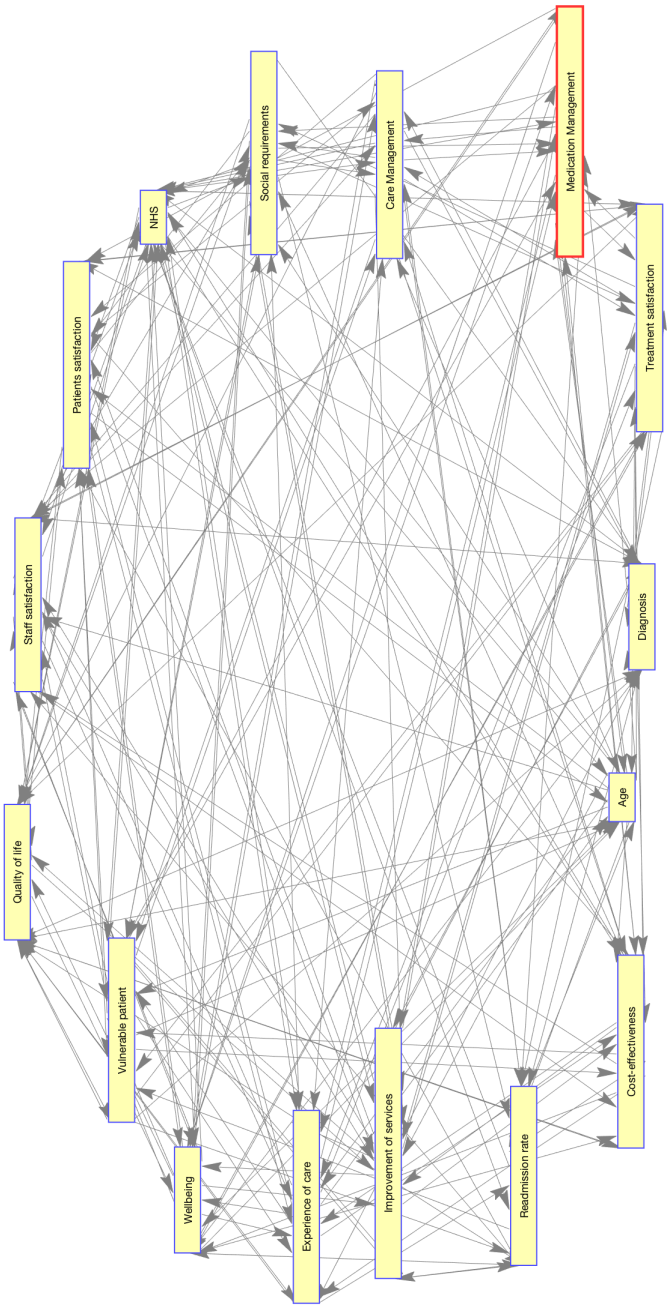

Figure 4.4.1: The relational dependency network as discussed in Section 2.10. [97]

Figure 4.4.1 depicts the complete network associated with the above concepts were the edges represent a dependency relation. Subsequently, using the method introduced in [116], we identified the Bayesian network depicted in Figure 4.4.2.

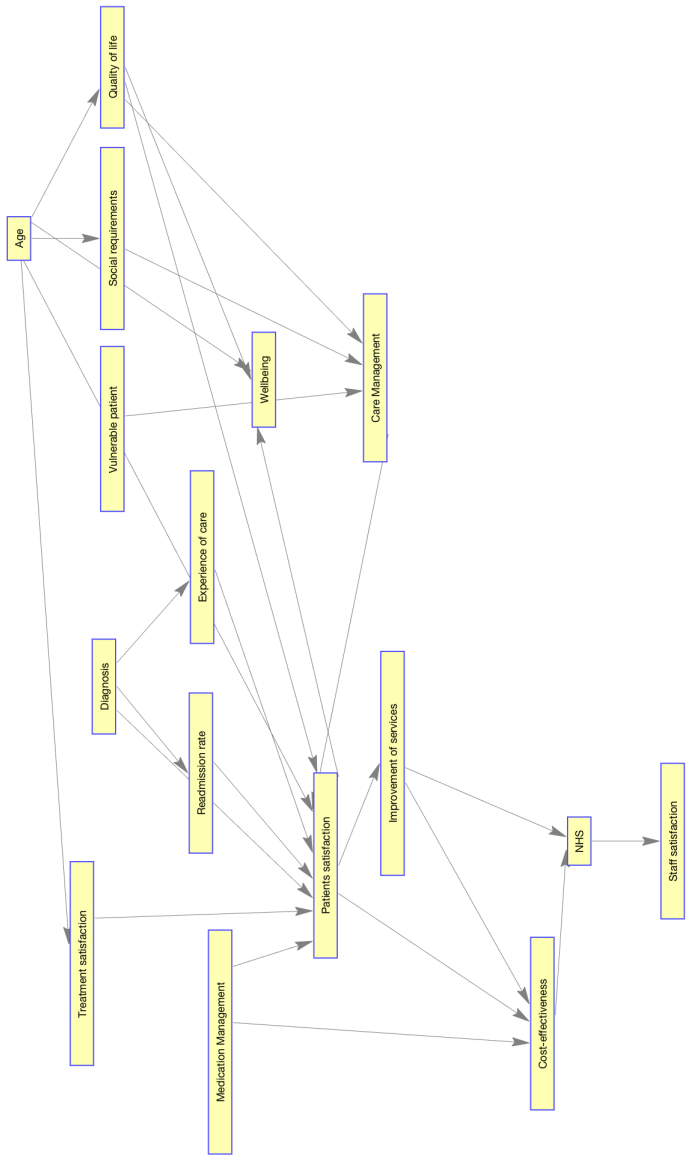

Figure 4.4.2: The BN extracted from the dependency network depicted in Figure 4.4.1. [97]

The assessment of the conditional probabilities between the concept above as in [116], which identified the following pairs of concepts, which exhibit a strong mutual influence:

- *Patient's satisfaction – Age*
- *Patient's satisfaction – Quality of life*, and
- *Patient's satisfaction – Readmission rate*

More specifically, the conditional probabilities are as follows:

- $P(\text{Patient's satisfaction}|\text{Age}) = 0.73$
- $P(\text{Patient's satisfaction}|\text{Quality of life}) = 0.59$
- $P(\text{Patient's satisfaction}|\text{Readmission rate}) = 0.61$

As discussed above, approximately 3000 tweets were analysed, which were selected based on the following keywords:

- Patient's satisfaction
- Hospital
- NHS

Figure 4.4.3 depicts the polarity based on positive, negative, neutral keywords, as well as their compound (overall) value for each tweet.

More specifically, we found the following values:

- Positive polarity: mean $= 0.076$ standard deviation $= 0.12$
- Negative polarity: mean $= 0.13$ standard deviation $= 0.15$
- Neutral polarity: mean $= 0.79$ standard deviation $= 0.16$
- Compound (overall) polarity: mean $= -0.1$ standard deviation $= 0.48$

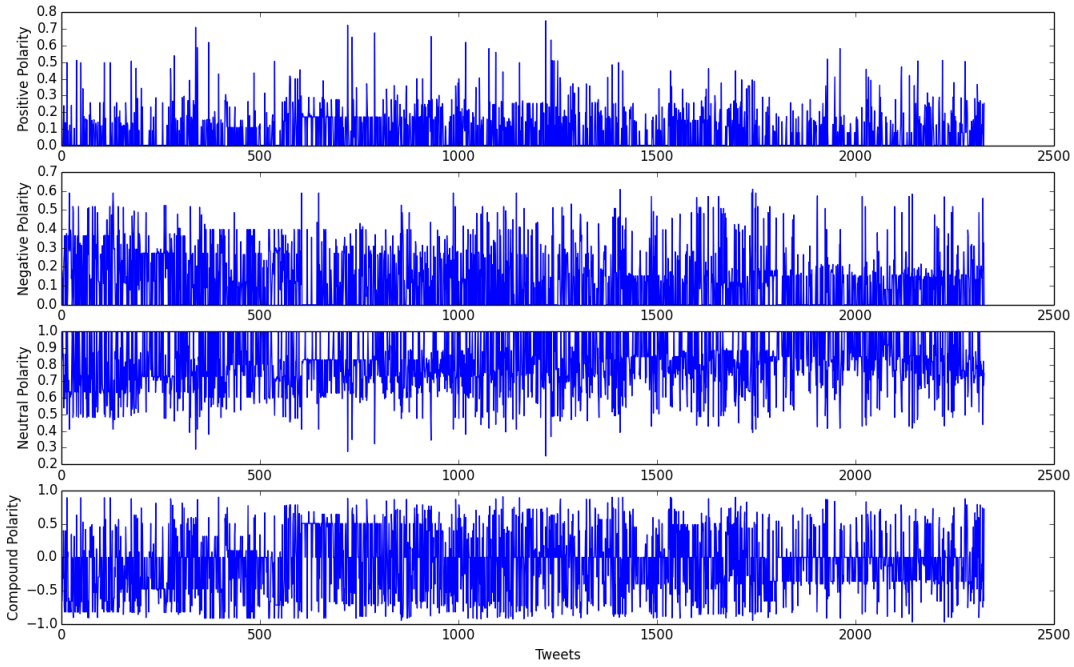

Figure 4.4.3: Polarity values of the tweets extracted as described in Section 2.10.[97]

Furthermore, a calculation of the correlation coefficient between positive polarity and overall polarity, and negative polarity and overall polarity, which gave 0.72 and -0.81, respectively. This indicates a strong positive and negative correlation between these pairs of concepts, respectively.

The above suggests that the overall polarity is close to being neutral with a significant fluctuation measured by the corresponding standard deviation. However, such tweets were identified just after the terrorist attack in Manchester, and as a consequence many tweets contained negative sentiment related to such event.

Combining the above two analyses, *Age, Quality of life,* and *Readmission rate* play an important role in patient satisfaction, and it appears that different age ranges are particularly significant. Even though the overall polarity is close to neutral and closely linked to positive polarity values, this can only be applied to social media users, which usually does not include older patients. Furthermore, considering the recent terrorist attack in Manchester, this suggests that negative polarity val-

ues may have been affected. Overall, the closeness of the overall polarity to a neutral level (represented in this case by 0) and statistical measures discussed above, suggest that the mood is fluctuating and it is directly linked with positive polarity values.

4.5 Summary

In this chapter, we have discussed a preliminary approach to determine and assess the contributory factors in patient satisfaction, based on an automated knowledge extraction method from articles from PubMed, as well from Twitter feeds. The proof of concept code and implementation developed to test this code are available in appendix chapters of this thesis.

5

Proposed Solution

The purpose of this chapter is to discuss the processes and rationale behind the need for a new type of data analytics platform. It discusses the initial stages of the proposed solution. It demonstrates an architectural design diagram and gives detail on each component.

The creation and availability of large datasets, has changed how businesses, scientific and academic fields design their experiments, collect data and ultimately, how usable insights are extracted. As a consequence, over the last decade, many data analytics environments and platforms have been proposed to enhance the decision-making process. In particular, real-time machine learning combined with detailed and interactive visualisations are likely to enhance the users' ability to process and create actionable outcomes from data.
However, the majority of the proposed solutions have serious limitations, which

only allow information extraction and analysis, under specific conditions. Despite the extensive advertising campaigns carried out by businesses, R&D organisations, as well as academic institutions, a comprehensive platform, which utilises a multi-disciplinary approach has not yet been developed. The way data is gathered, analysed, visualised and adjusted according to the users' needs depends on numerous parameters, which are rooted in various disciplines and approaches.

5.1 Proposal of Integrated Solution

As discussed above, one of the most recurrent issues found across the current market offerings is the lack of full integration of visualisation and interactivity capabilities, as well as exhaustive data analysis. As part of the current research efforts, this thesis demonstrates a novel data analytics tool, which combines suitable features to address the above challenges.
In this section, we will briefly describe part of the general architecture, which aims to create a dynamic, responsive and accurate tool to enhance the decision-making process. The main components of the proposed approach are outlined in Figure 5.1.1. In particular, this shows how the interactivity between individual components is key to a useful system that provides business advantage over existing solutions. An outline of each component is discussed below:

- Keyword Identification: this process focuses on the assessment and definition of specific keywords if textual input data sources are available. This will be used to refine the searching and trawling component by specifying a context or scenario, which will enable a greater precision.

- Text extraction: the relevant data is mined and assessed to identify the relevant information, based upon search terms. This identifies what text and metadata will be included in subsequent analysis.

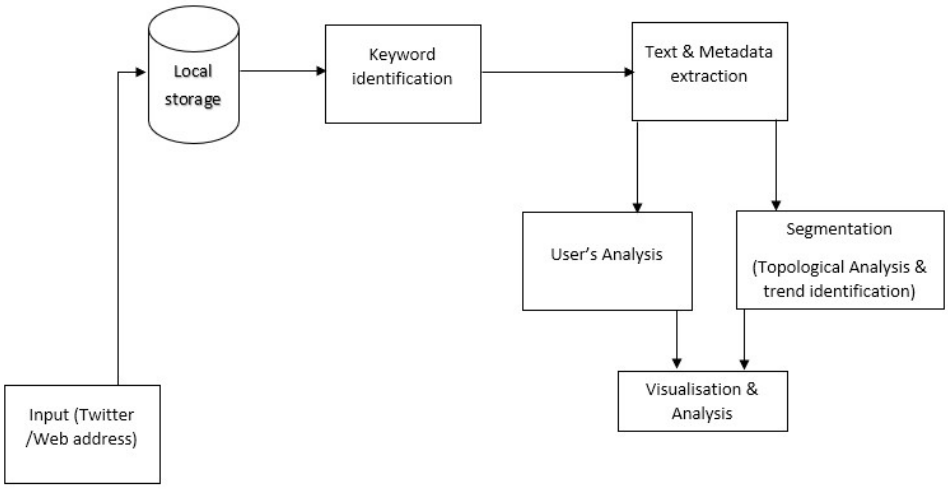

Figure 5.1.1: Component connectivity & systems flow of the proposed solution. [96]

- User's analysis: during this stage, the user can interact with the system via a UI. Specific information will be used to gather information regarding properties and attributes of concepts, topics and products that are used to guide the analysis.
- Web trawling and sentiment analysis: general and more specific websites are identified based on pre-defined keywords, as well as other search criteria. Social platforms will also be investigated.
- Visualisation: the results are visualised via an appropriate UI.
- Segmentation and analysis: during this process, the information retrieved and entered into the system will be used to identify and analyse specific segments of the data such as identification of specific trends and topic identification.

In particular, the ability to gather information across key global impact factors such as social media feeds and web forums or web sites can provide a vital insight into customers' opinions with regards to a product, service or brand. In particular,

the ability to monitor such feeds in an automated but reliable manner is likely to create valuable direction for product development. Sentiment analysis has developed significantly over the past 10 years and will continue to become more accurate while requiring fewer resources as new techniques are introduced [72, 120]. Any data analytics system must be able to quickly embrace these changes to maintain a cutting-edge position as an industry leader in the implementation of machine learning for business intelligence.

In [41], the authors emphasise that an efficient approach must capture the audience's attention and help them to engage with the data so that the overall understanding is increased. Furthermore, when data is accessible, users can better understand how a machine learning algorithm has reached its final decision. The level of trust and the value people perceive from machine derived decisions is expected to increase, especially if the machine learning algorithms will lead to correct decisions and predictions [68].

Another important component is the segmentation and analysis elements. Whilst it is entirely possible for the machine learning to identify and classify what segment of the data fits the purpose without a visualisation, this could lead to a lack of understanding and subsequent rejection of the analysis carried out. However, requiring continuous and not optimised human interaction will lead to a slower and more costly decision-making process. The segmentation and analysis components are part of a set of tools, which have been developed to integrate novel algorithms based on network theory, topological data analysis, text mining techniques and automated extraction of decision models [116, 120–122]. These are part of an IP process and as a consequence, at the moment they cannot be described in details.

5.2 Summary

The outline discussed in this chapter aims to provide a unified solution of interactive visualisations that help shape the automatic decision-making process and give

greater insight to the user as to what parameters have been used to influence the overall decision. The tool is aimed at those within the healthcare environment tasked with improve patient experience. This aspect will aid with the usability of advanced machine learning, whilst providing easy access to powerful machine learning tools to enhance the business decision-making process. This system integrates machine learning and topological data analysis creating a novel method of data analysis when used for social media text analysis.

6

Results & Prototype Validation

This chapter discusses the approaches taken and demonstrates proof of concept code for each element of the proposed system. The prototype and its development are discussed along with details of how elements of UI design and code principles have shaped the construction of the application. Two versions of sentiment analysis techniques have been prototyped and discussed. The chapter discusses and demonstrates how the neural network text classification takes place along with website keyword extraction, the final part of the chapter deals with visualisation techniques and integration of the application with suitable web frameworks.

Production of a near-real-time machine learning system that requires no GPU support for machine learning tasks is the primary goal of this prototype. The system should be capable of providing a close to real-time output from social platforms. The system will be produced for a computing environment that does not have a

machine learning specialised GPU to speed up computation. The initial prototype will be required to move across many systems including Windows and Mac OSX, the prototype must be platform-agnostic, this does create some limitations with deployment methods, Python does not offer reliable run time build solution such as .exe, as such the scripts that are created need to be used directly to start the application, therefore all test systems must install a python 3 environment. The final completed build is proposed to be a web-based front end, allowing for updates and changes to the product to take place centrally within an organisation the end-users do not experience any update or alteration to work patterns. The use of a web client allows users to utilise the machine learning service on any device and maintain an uninterrupted workflow across any operating platform, including mobile. A web front end of Django connected to a database system and python back-end to allow the machine learning and data analysis to take place. For this prototype to test the interoperability of gathering streaming social inputs and web-based content, a simple to use Tkinter / TTK Python 3 GUI is the preferred option. This chapter discusses the approaches taken and demonstrates proof of concept code for each element of the proposed system.

6.1 Prototype

The prototype has been developed in Python 3, GUI elements have been constructed using TKinter and TTK. NLTK, genism, pyLDAvis, sklearn and kmapper library's have been used for aspects of the machine learning, the neural network text classification algorithm has been manually coded in python. Each aspect of the system has been developed in a stand-alone application for testing with a subsection of this chapter dedicated to each prototype, the final GUI design has not been designed, however, a multi-aspect prototype, with simple GUI has been developed and is in use with an external client for testing and further development of the social media discovery (not hospital-related) tools, some testing of web frameworks suitable to host such an application have been tested and are outlined at the end of this chapter 6.7.

6.2 Sentiment Analysis

Two approaches for sentiment analysis have been prototyped, first, the coreNLP approach from Stanford [76]using its stand-alone server architecture and second the VADER social sentiment library. An attempt to modularise both sentiment tools was unsuccessful with only the VADER sentiment library allowing for a 'one-click' application.

6.2.1 coreNLP

To test coreNLP, a stand-alone sentiment analysis test with an implementation of Stanfords coreNLP [76] using the Stanford CoreNLP server and Brat visualisation. A twitter feed search "Donald Trump" was stored into a flat-file database and then manually entered into the Core Server web interface, the output generated by figure 6.2.1 demonstrates the effectiveness of the solution, as already proven [10, 91] the CoreNLP suite produces highly accurate and reliable results. The CoreNLP solution requires a CoreNLP server to be configured and hosted on the local device, this process creates a significant barrier in usage not only for testing but for deployment. The python implementation of CoreNLP and Brat visualisation allows a direct comparison of each input on a classification scale represented by changing colour, whilst this results in visually appealing outputs it requires a more detailed and complex code solution to manage the outputs for further analysis.

Figure 6.2.1: Output from Sentiment analysis and Brat visualisation from StanfordNLTK processing.

The ability to package the CoreNLP server into a small python file proved unsuccessful, whilst it is possible to develop scripts that link via API to the CoreNLP server. The solution requires a CoreNLP server to be configured and active. The overall time taken to configure this server, and the resources required to maintain a CoreNLP server did not warrant the development time. Alternative methods and libraries have proven faster and easier to deploy for initial prototyping.

6.2.2 VADER sentiment in NLTK

The second solution for sentiment analysis involved the real-time streaming of a twitter search term into sentiment analysis library VADER (Valence Aware Dictionary and sEntiment Reasoner): A parsimonious rule-based model for sentiment analysis of social media text. VADER uses a sentiment lexicon that contains intensity measures for each word based on human-annotated labels. VADER was designed with a focus on social media texts making it an ideal starting point for this project. VADER places a lot of emphasis on rules that capture the essence of text typically seen on social media, for example, short sentences with emojis, repetitive vocabulary and copious use of punctuation (such as exclamation marks). VADER performance (F1 score) when tested with sentiment tasks for social media tweets the exact purpose the system had been developed for produces 99.04% Accuracy resulting in an 84.29% macro-F1 score [100]. The implementation allows for customisation and automated input of the text. Performance per tweet is good, and allows for near real-time analysis, its rule-based approach does capture a good amount of fine-gradation and few cases that are truly negative get classified as positive, and vice versa.

The prototype VADER implementation and twitter live streaming method are detailed in the appendix.

The code generates two outputs, first, a graph 6.2.2 that plots the number of tweets and the associated sentiment, this graph demonstrates sentiment over time as a linear series, giving a general picture of sentiment for a set of given search

terms. For demonstration purposes, the line graph was used as it easily demonstrates change over time. However, dependent on what level of granularity a Bar chart could be used to combine all positive sentiment and all negative sentiment showing a general, or a large chart demonstrating all values in 0.1 increments to show details on how severe the sentiment is for the given term.

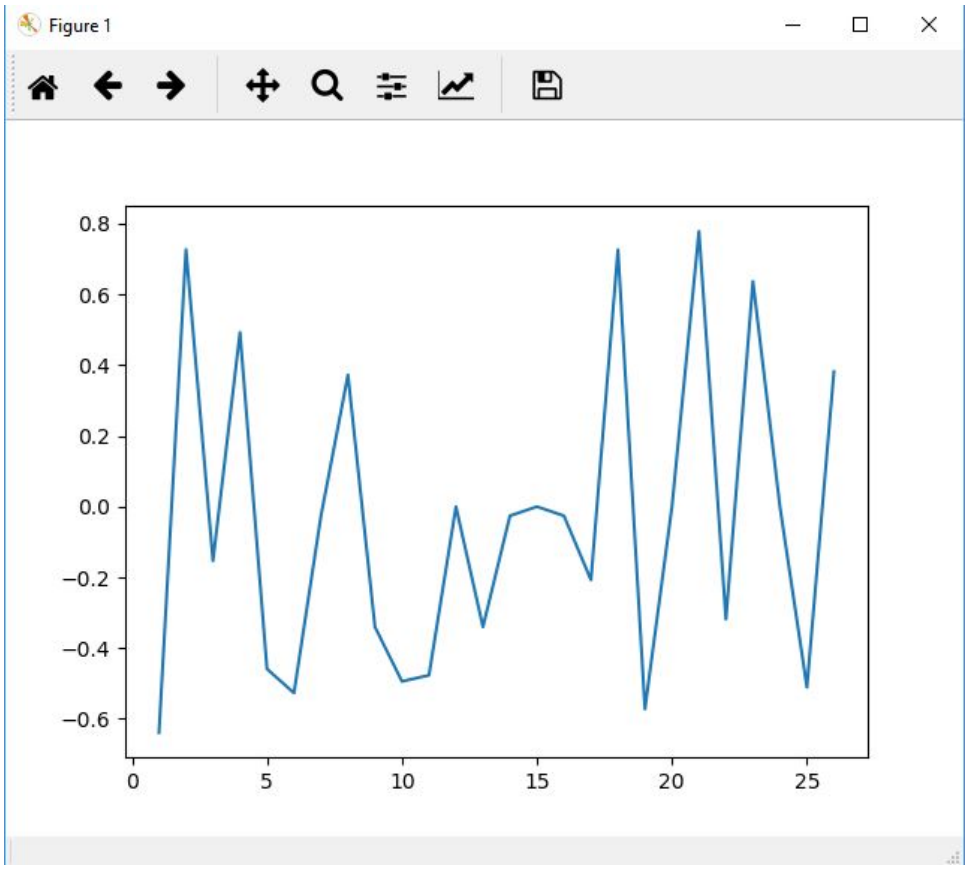

Figure 6.2.2: A graph generated from live sentiment analysis y = Sentiment intensisty, x = Number of inputs

The second output is a textual output that breaks down each tweet and demonstrates the part of speech tag that has been identified along with the sentiment likelihood, it is expressed in 4 categories, 'neg' – negative sentiment, 'neu' – neutral sentiment, 'pos' – positive sentiment and finally 'compound' – the overall score

given to the input. The values for neg, neu and pos range from 0 to 1 whilst the compound score ranges from -1 to +1, a compound value that is negative (-) shows that the overall sentiment of the given input is likely to be negative, with the value a score indicating the severity of the categorisation.

```
RT @DavidJHarrisJr: SAD!!
Another Hollywood liberal saying Trump supporters who are
    black have a "mental illness" while white Trump …suppo
[('RT', 'NNP'), ('@', 'NNP'), ('DavidJHarrisJr', 'NNP'),
    (':', ':'), ('SAD', 'NN'), ('!', '.'), ('!', '.'), ('
    Another', 'DT'), ('Hollywood', 'NNP'), ('liberal', 'JJ')
    , ('saying', 'VBG'), ('Trump', 'NNP'), ('supporters', '
    NNS'), ('who', 'WP'), ('are', 'VBP'), ('black', 'JJ'),
    ('have', 'VBP'), ('a', 'DT'), ('``', '``'), ('mental', '
    JJ'), ('illness', 'NN'), ("''", "''"), ('while', 'IN'),
    ('white', 'JJ'), ('Trump', 'NNP'), ('…suppo', 'NN')]
{'neg': 0.274, 'neu': 0.615, 'pos': 0.111, 'compound':
    -0.639}

RT @TaxReformExpert: President Trump works 24 hours a day
    for 365 days a year for free.'
Wouldnt it be nice if the media gave him the …appr
[('RT', 'NNP'), ('@', 'NNP'), ('TaxReformExpert', 'NNP'),
    (':', ':'), ('President', 'NNP'), ('Trump', 'NNP'), ('
    works', 'VBZ'), ('24', 'CD'), ('hours', 'NNS'), ('a', '
    DT'), ('day', 'NN'), ('for', 'IN'), ('365', 'CD'), ('
    days', 'NNS'), ('a', 'DT'), ('year', 'NN'), ('for', 'IN
    '), ('free', 'JJ'), ('.', '.'), ('Wouldn', 'NNP'), ''('',
    'NNP'), ('t', 'VBD'), ('it', 'PRP'), ('be', 'VB'), ('
    nice', 'JJ'), ('if', 'IN'), ('the', 'DT'), ('media', '
    NNS'), ('gave', 'VBD'), ('him', 'PRP'), ('the', 'DT'),
    ('…appr', 'NN')]
{'neg': 0.0, 'neu': 0.79, 'pos': 0.21, 'compound': 0.7269}

RT @glennkirschner2: I 'didnt want to call McConnell an
    "enemy of the people". But because he single-handedly
    blocks the legislative …proces
```

```
[('RT', 'NNP'), ('@', 'NNP'), ('glennkirschner2', 'NN'),
   (':', ':'), ('I', 'PRP'), ('didn', 'VBP'), ’('', 'JJ'),
   ('t', 'NN'), ('want', 'VBP'), ('to', 'TO'), ('call', 'VB
   '), ('McConnell', 'NNP'), ('an', 'DT'), "('', 'NNP'), ('
   enemy', 'NN'), ('of', 'IN'), ('the', 'DT'), ('people.',
   'NN'), "('', 'NN'), ('But', 'CC'), ('because', 'IN'), ('
   he', 'PRP'), ('single-handedly', 'RB'), ('blocks', 'VBZ
   '), ('the', 'DT'), ('legislative', 'JJ'), ('…proces', 'NN
   ')]
{'neg': 0.09, 'neu': 0.849, 'pos': 0.061, 'compound':
   -0.1531}

RT @miamijj48: Michael Shannon Tells Trump Supporters ’Its
   Their Time to D!E Now LiberalPrivilege ""Elites  https://
   t.co/qmH0apiBxs
[('RT', 'NNP'), ('@', 'NNP'), ('miamijj48', 'NN'), (':',
   ':'), ('Michael', 'NNP'), ('Shannon', 'NNP'), ('Tells',
   'NNP'), ('Trump', 'NNP'), ('Supporters', 'NNP'), ('It',
   'PRP'), ’('', 'VBD'), ('s', 'VB'), ('Their', 'PRP$'), ('
   Time', 'NN'), ('to', 'TO'), ('D', 'VB'), ('!', '.'), ('E
   ', 'NNP'), ('Now', 'RB'), ('#', '#'), ('LiberalPrivilege
   ', 'NNP'), "('', 'NNP'), ('Elites', 'NNP'), "('', 'NNP'),
    ('https', 'NN'), (':', ':'), ('//t.co/qmH0apiBxs', 'NN
   ')]
{'neg': 0.0, 'neu': 0.825, 'pos': 0.175, 'compound': 0.4926}

RT @ChrisWick__: I would like to emphasize to everybody that
    Eric Trump's foundation was actually seen funneling
   cancer cells study cash …st
[('RT', 'NNP'), ('@', 'NNP'), ('ChrisWick__', 'NNP'), (':',
   ':'), ('I', 'PRP'), ('would', 'MD'), ('like', 'VB'), ('
   to', 'TO'), ('emphasize', 'VB'), ('to', 'TO'), ('
   everybody', 'VB'), ('that', 'DT'), ('Eric', 'NNP'), ('
   Trump', 'NNP'), ("'s", 'POS'), ('foundation', 'NN'), ('
   was', 'VBD'), ('actually', 'RB'), ('seen', 'VBN'), ('
   funneling', 'VBG'), ('cancer', 'NN'), ('cells', 'NNS'),
   ('study', 'VBP'), ('cash', 'NN'), ('…st', 'NN')]
```

```
{'neg': 0.212, 'neu': 0.692, 'pos': 0.096, 'compound':
    -0.4588}
```

The ease of implementation, speed and accuracy of the VADER solution allow the output to be directly analysed and a real-time graph produced that gives immediate feedback on sentiment. The choice of the graph can be easily modified as the output data from the VADER function is easily accessible. The VADER function, when combined with real-time analytics, in this instance graph drawing, does not impact system performance and only adds a relatively small additional processing, the additional time taken is almost unnoticeable. This method is an ideal starting point and the proof created here demonstrates how existing technologies are ready to be utilised in less theoretical scenarios as usually researched by academics.

6.3 Neural network Text classification

To process the data into meaningful and actionable outcomes, the input data requires more processing than just sentiment analysis. Text classification and topic modelling can provide an insight to hospital administrators about issues arising and general topics patients are concerned about within their hospital. The prototype method used involves a 2 layer neural network, with the initial test performed by hand-coding classification groups 7.4.1. The network input comes from a bag of words created via lancasterStemmer from the tweets, this is input into the first neural network layer and activated via a sigmoid function. The visualised workflow of the machine learning 6.3.1 gives an overview of how 7.4.1 performs at each stage.

The Input layer for the neural network is created by NLTK tokenize followed by Stemming and finally converting to Bag of words that is generated from the social feed or any textual source. The hidden layers are activated with a Sigmoid function, whilst ReLu could be substituted here, for code simplicity and ease of implementation sigmoid has been used, future use of PyTorch 7.4.1 will allow much simpler codebase allowing changing the activation functions or neural network hy-

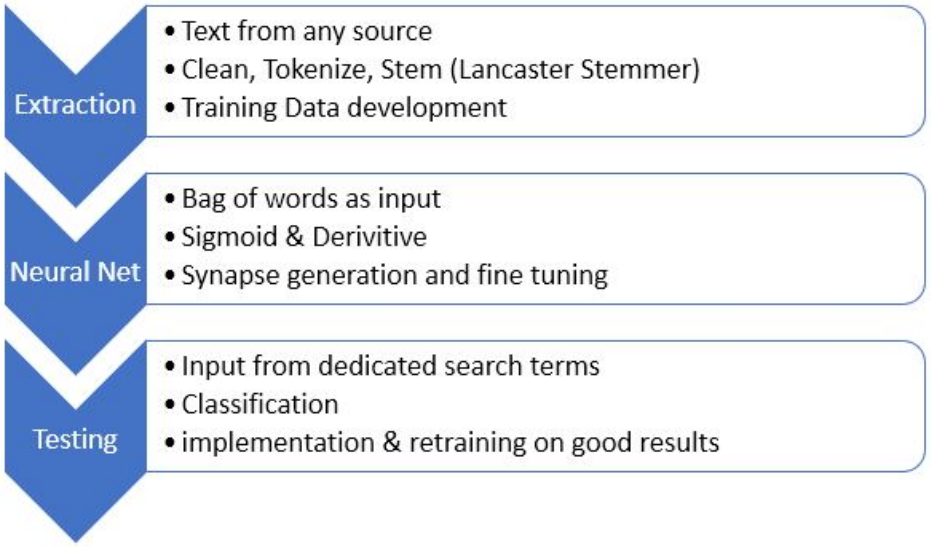

Figure 6.3.1: Flow chart demonstrating the process

perparameters. The initial seed for the neural network synapse weights has been produced via the Python random number generation function, whilst this is not the greatest random number generation method it is more than adequate for this function. The code allows for fine-tuning of the system via variable manipulation of hidden neurons, alpha and epochs. However, such tuning will potentially reduce the ability of the system to cope with the current wide range of topics, the fine-tuning will improve results on a specific training set, but is likely to reduce the ability of the system to cope with any textual based set of results.

In the prototype, the training data is created from a very small set of tweets between 3 – 4 examples for each category. For machine learning, it is uncommon to use such a small amount of training data. However, even with this limited training set, it is still a capable categorisation system. The simple output below demonstrates the capability of the neural network with three examples.

```
1 classify("The charity fundraising event is tomorrow at
    @Alderhey")
2 classify("My mate is doing a charity hike for alder hey, any
    donations greatly appreciated.")
```

```
classify("The nurse has been such a great help today")

The charity fundraising event is tomorrow at @Alderhey
classification: [['AlderHey_event', 0.9286088266363973]]
My mate is doing a charity hike for alder hey, any donations
    greatly appreciated.
classification: [['AlderHey_thank', 0.37539888396811094]]
The nurse has been such a great help today
classification: [['staff', 0.6802888005247033]]
```

The test case and training data are taken directly from twitter with no alterations or manipulation, the ease and simplicity of this method will allow for users to train and improve the system with an approval system, manual classification of tweets by a user is simple to demonstrate and easy for a user to perform. The results whilst not perfect, clearly show that such a simple method can yield surprisingly accurate results. The lack of training data will be overcome with the simplicity of adding new training cases, allowing users to update the model appears to be a viable solution.

6.4 Visualisation of network groups

The results and output from the neural network can easily be manipulated and analysed, in 6.4 methods of data visualisation based upon this neural network have been prototyped. The neural network prototype has highlighted some benefits of this approach, the neural network approach to classification and grouping is not biased towards a particular data set, it has not been developed to excel in a particular area unlike VADER and social sentiment. The acceptance of universal input such as word documents, social feeds, emails, job descriptions, Family and friends test answers, or NHS reviews data enables end-users to discover trends and noteworthy topics across a much broader scope, with such large amounts of data a system capable of analysing a significant portion has benefits over a system designed to only function on one particular media type. This approach, however, is not without its own unique set of limitations, the approach is supervised and requires careful consideration of the data used to train the network. The super-

vised approach requires sufficient domain knowledge to create meaningful groups that will be used for training the classification groups. The data visualisation stage allows for complex data analytics to take place but enables rapid comprehension of such complex analysis, implementation of two methods discussed in chapter 4 have been prototyped one created with TSNE and wordvector space7.4.1, the other with LDA(Latent Dirichlet Allocation) 7.4.1.

The TSNE code requires that the number of topics to be discovered be set at the input, this requires a trial and error situation for discovering at what point topics are no longer useful, this manual process must be changed for each data set that is to be evaluated, in the code 7.4.1 a flat-file data source has been used. The data library did not support on the fly rendering of the graph.

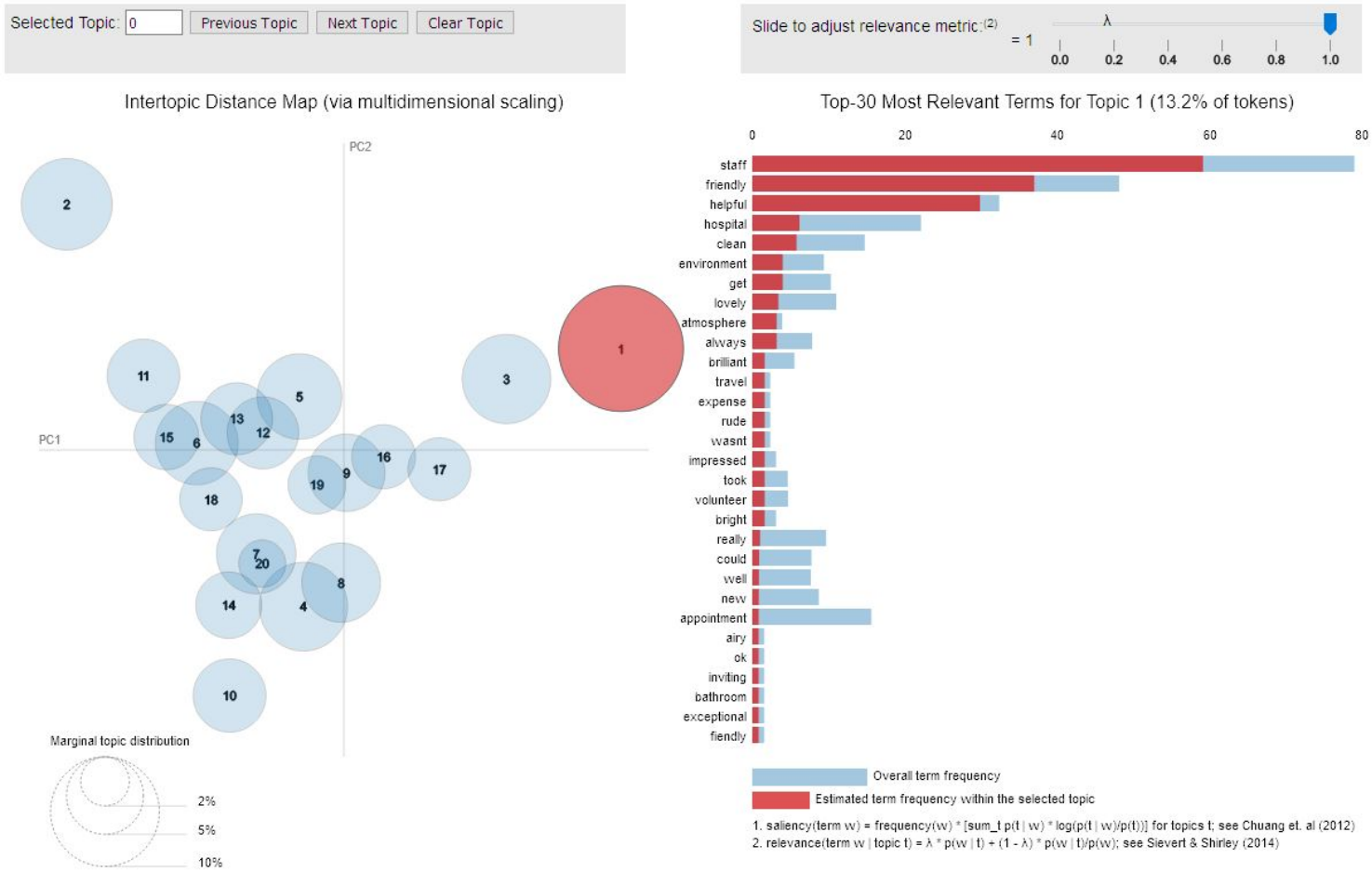

Figure 6.4.1: Intertopic Distance Map

An unsupervised approach has not been prototyped but could allow for automatic topic discovery to occur without explicit topic values to be set or creation of pre-set categories, a more hands-off approach. The creation of autogenerated topics and topic discovery could be implemented to provide an overview of discussions taking place on social platforms or topics generated by family and friends test, LDA vis used on the friends and family test data gave topic overviews, whilst not real-time it is fast and more efficient than a manual review process. This current prototype would not be capable of switching to an unsupervised approach therefore a new algorithm and method would be required.

The LDA(Latent Dirichlet Allocation) mapping method (Code available in appendix) provides an initial topic overview and allows visualisation of the total number of members in each topic and does this in a visually appealing nature. That can help increase interest in the data set. The distance map via TSNE word vector provides more information, with key terms easily visible at a glance, more investigation is required into how the data provided in the TSNE wordvector visualisation and the grouping provided by Latent Dirichlet Allocation can give users the network overview and help visualise the connections that are provided via Latent Dirichlet Allocation and the deep interrogation of TSNE.

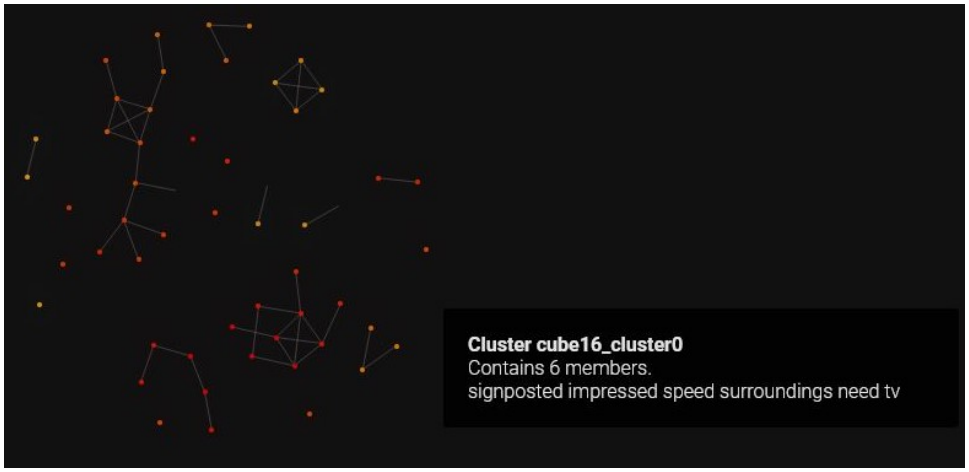

Figure 6.4.2: Topic network map from Family and Friends dataset

The Latent Dirichlet Allocation network map 6.4.2 allows interactivity with the nodes and is visually appealing, however detailed data retrieval is not ideal with this method, mouse rollover highlights are active on the visualisation but immediate understanding of the data set is limited to group size and intensity. This visualisation can be useful but, as a sole means for data analysis, it would not provide significant added value to the data.

6.5 Web Keyword Extraction

The ability to extract data from other sources and build similar topic recognition and sentiment analysis can also be performed, the system utilised in this prototype gathers the most popular webpages when a given search term is queried via google and parses each page to create a per web page corpus, this corpus consists of nouns and adjectives, the pages are then aggregated and a combined result displayed via a bar chart 6.5.1. The code available in the appendix 7.4.1 demonstrates how standard web libraries have been adapted to provide input and then create a topic analysis for the search terms.

Extraction from the top 10 websites can be customised by expansion or supplemented with a selection of pre-chosen websites, for hospital data the inclusion of NHS reviews or the NHS health A-Z can help supplement topics with known quality resources that can help guide topic creation based upon trending keywords. Discovery of related websites and terms from a Google search can assist researchers in discovering new and rising trends, that may influence hospital attendance. The ability to quickly condense a large range of web content greatly simplifies and reduces the time taken for a researcher to assist with managing current trends that are bringing patients into A&E. The ability to combine social trends with web trends will allow the hospital to pre-emptively develop information packs and plans that address the current rising issues. Data representation can be modified with any of the existing python data visualisation techniques, as the file is not required to be updated on a real-time basis, the run once nature of this analysis

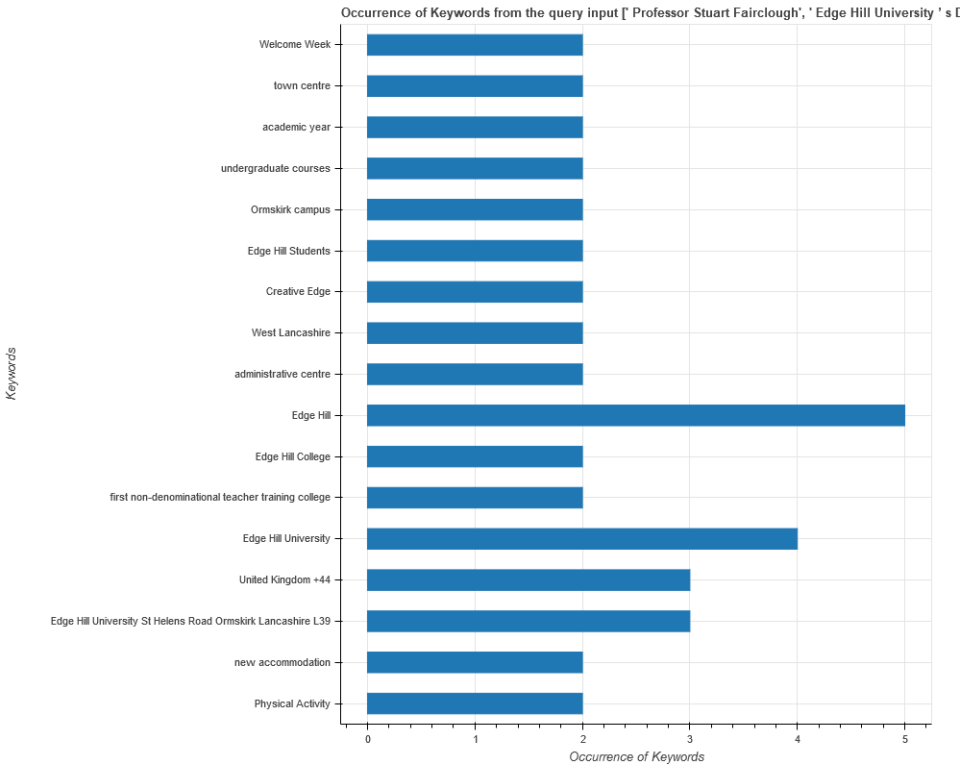

Figure 6.5.1: Bar Graph of extracted terms based upon keyword web relevance search for Edge Hill University

means that any method of producing a graph or chart will be suitable for this visualisation. This allows any of the available graphics library's available in python to be used when generating this visualisation.

6.6 Prototype GUI

The ability to compile and run source code is deemed a trivial task for a computer programmer, however, none computer specialists have no experience or knowledge on how this process is performed or how to go out configuring a development environment suitable to running a machine learning and analysis tool. One of the main priorities of this project is to bring the advancements in machine learning to a more mass-market audience so that the benefits of advanced analytics can be

utilised by healthcare professionals to benefit the general public. To allow this the application and prototypes require a graphical user interface (GUI). Various options are available for creating a GUI in python, the GUI prototype has been built with TKinter library along with the TTK helper library.

The code (see appendix 7.4.1) demonstrates how the existing prototype modules have been integrated with a simple but effective GUI 6.6.1, users can easily utilise each function without the need for a command-line or python enabled programming environment. The code implements the data feed and sample data to supplement the outcome ensuring enough data points have been reached to create output visualisations. The TTK and Tkinter approach does not look as visually appealing as some applications that exist today but is only intended to prove functionality for a GUI front end. Customisation of each element could be achieved at this stage with work from a graphic designer who could produce individual buttons and create an accessible colour scheme. Whilst the TTK interface removes the burden of command-line interfaces from the user it still requires a python environment to be configured along with the relevant libraries installed. Whilst it is possible to compile python code into an exe this approach is not recommended or encouraged. Rather utilising the bundled PIP (python installer package) command and requirements .txt to detail the required libraries and the correct version numbers will automate the download process and allow the GUI and python script to function in a very similar fashion to a standard executable.

6.7 UI Development

To create a seamless experience for the users the burden of installing a python environment should be removed and a transition from a stand-alone client to client server-based environment would be best. To create a powerful user interface that not only performs the required machine learning but also is easy to use and visually appealing. Options that have been initially explored include Django with HTML, Django with java & NPM, Django & docker. To successfully link a web

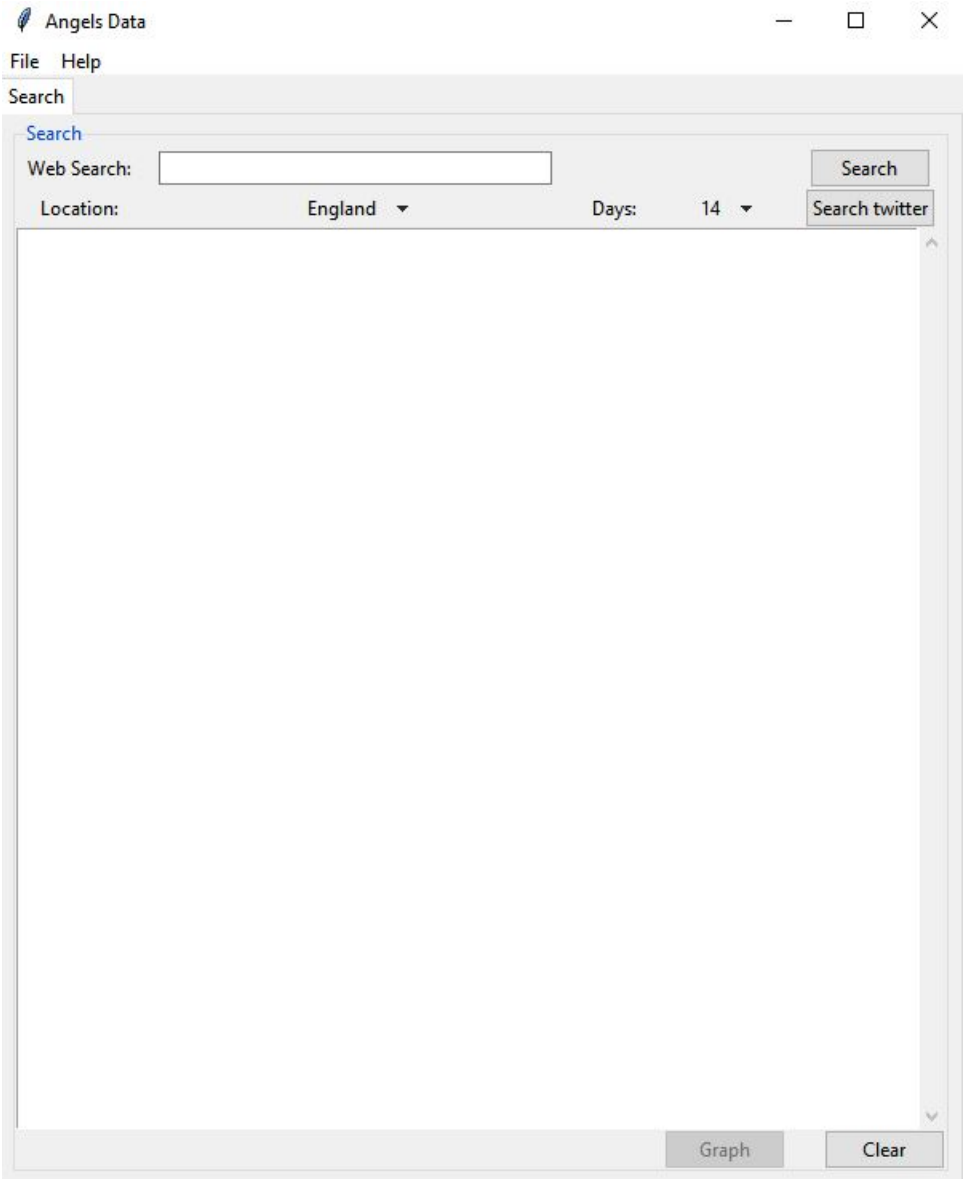

Figure 6.6.1: Prototype TTK UI - Main input page

front end to a python machine learning implementation some form of API or a direct link is required, two of these options are Django and Flask. For the prototype a Django implementation was settled upon due to its completeness in a one package approach, that is similar to the ethos of this project, a self-contained en-

vironment. The Django framework allows direct connection to a python backend (script) and provides a rendering method for HTML pages or via some configuration JavaScript. The JavaScript approach became bloated and unmanageable with changes requiring expertise in many web technologies and programming languages that to create a suitable prototype would take a significant development.

6.7.1 Django with JavaScript

The Django and JavaScript approach allows for a rich web experience with numerous available resources and quality functionality that has already been produced for interactive web pages and web applications, the benefits of using JavaScript for this approach allow rapid development of the GUI and front-end architecture. However, creating a link between JavaScript on the front end and python back-end requires multiple intermediate stages, the creation of a REST API is one solution. The REST API allows communication between the user interface and the Django SQL database, as this rest API is custom its functionality can be created to directly support the process. To create a full API can be time-consuming and introduces security risks for data access on the database. To maintain three separate foundation technologies for use in one application is overly burdensome for a single researcher. The JavaScript and NPM environment requires setup and a server, the Django configuration and finally, the Machine learning (NLTK) requires expertise in each of these areas to ensure secure implementation of each technology.

6.7.2 Django with Docker

A potential solution to the large configuration issue caused by three foundation technologies is to use Docker, the docker system creates a stable environment that allows for deployable configurations, a stable and working app can be created and then bundled into a docker environment which would then be deployed at the target locations. This approach will allow the use of many more technologies but requires more expertise in the use of docker as a deployment method.

6.7.3 Django with HTML

A solution that has had some prototyping is the use of Django and HTML, the prototype has enabled a visually pleasing interface and a central repository for the system to be hosted. The web page interface allows any device to access the system. A single point of maintenance and high powered system to perform the machine learning can aid with its processing speed enabling the close to real-time analysis required to process a social media feed to monitor for trending topics that may influence hospital admissions.

6.8 Summary

Based upon the initial testing and development in machine learning since the project began in 2017 a system as described above is highly capable of performing machine learning on textual sources. The given solutions can manage a social media stream or textual web content and providing a valuable output in the form of categorisation and grouping. The systems adaptability is a key factor that proves how easily and capable a full environment would perform. A downside to this python approach as it is prototyped is the python environment required to run the system, the additional python libraries are small excerpt for the NLTK corpus, A switch to Pytorch could significantly reduce this reliance upon large corpora, reducing the disk space and allowing for even more portability should the application remain a client only application. A move towards a client-server model would allow for an array of methods to be implemented from a pure neural network system alongside the prototyped hybrid corpora and neural network approach. The prototype UI and all modules are being tested by a third-party company who to date have provided positive feedback regarding performance and usability. Initial testing and reports from the company have highlighted the importance of managing the training set and providing suitable user training in how to prepare the training data set along with the process of updating the model. The results varied in neural network success as the training data quality shifted, with more detailed information

provided to end-users on the manual process of data selection this problem can be remedied.

7

Conclusion

This research focused on using open source machine learning to create a data analytics tool, focusing on a tool that can be used without any knowledge of machine learning or computer programming. Making the system and theoretical / academically proven machine learning concepts completely accessible. The research has demonstrated how open source machine learning and data visualisations can improve data gathering and data analytics. The solution and prototype described in this thesis add substantial proof that the currently available open-source solutions are capable of augmenting traditional methods of data discovery and provide a solid foundation that proves the suitability of machine learning for hospital data analytics. Hype created by large corporations has tainted public opinion of machine learning, making consumers wary of its benefits and consumers struggle to see where machine learning fits into everyday life, with computers such as IBM Watson performing on TV or Alpha Go, again, performing at competitive Go, con-

sumers and end-users are being misled as to the complexities and the portrayal that machine learning will take over rather than assisting and reduce burdensome tasks. Tasks such as reading thousands of tweets to assess if any of them mention the hospital or its staff and if so what message is being broadcast and to how big an audience. This research shows the machine decision as a score of certainty allowing end-users to make an informed decision. In addition, reducing the time taken to sift through a large data set.

7.1 Place within existing literature

This work adds to the literature that supports the positive effect social media has made upon the health industry. Combined with the positive impact being made by social platform interactions, this work adds the ability to analyse and understand social trends. This will further help develop a more rounded and beneficial approach to social media usage and analysis within the healthcare industry.

7.2 Contribution to Knowledge

In particular, the main contributions of this research include (but not limited to):

The thesis demonstrates how better integration of data extraction from various sources, such as blogs, social media and general web resources can be used to supplement the decision-making process.

The thesis contributes towards the application of machine learning in a business environment, via the integration of machine learning and topological data analysis. Transforming theoretical machine learning methods and with the creation of novel algorithms and methods creates an application of machine learning and text analysis in python.

The methods utilised here have not been widely integrated before this research. The thesis has demonstrated that given modern open-source libraries it is entirely possible to create machine learning systems that can utilise a low powered system (no GPU assistance) to create real-time machine learning assisted analytics from

social media as well as processing collected data from hospital sources.

We learn from chapter 1 that as data gathering and data storage has become more common, business are placing more emphasis on the power of data and data-driven analysis. The data-driven approach in this instance includes many research areas and special interest groups ranging from Network Theory, Bayesian Networks, NLP and Machine Learning. The ability to process this level of data is helping business enrich forecasting projections and help build actionable insights based upon the data collected. Chapter 2 discussed how best to determine contributory factors in patient satisfaction and how automated knowledge extraction can work in a medical context, with PubMed forming the basis of groups based upon textual content, the graphing of discovered topics allows for quick and insightful data to be gained from a large corpora of Pub Med articles that would take significant time to read and create suitable topics from each article. Chapter 3 identified the current process for adverse event identification within the health care environment and discovered that little to no process or policies are common for social media adverse event identification and management. Identification of important factors and a recommendation for improvements are suggested, the information in this chapter can be used as a starting point for discussions on how policy and identification of adverse events can be achieved. As social feeds become a more important means of interaction between an organisation and patient, the level of data generated will increase to levels unmanageable by an individual, machine learning will enable processing of this information creating a suitable platform for judging adverse events. It is clear from chapter 4 that whilst commercial machine learning suites do exist, they are highly specific to the vendor that produces the system, each software offering is highly specialised towards a particular industry and focused around numerical data, such as sales forecasting. A clear area for expansion exists where a small scale but powerful machine learning algorithm can perform small actions to assist in a wider analytical environment. From the options investigated in this chapter, the Anaconda Suite provided the greatest flexibility, allowing the use of machine learning to be built on lightweight and directed machine learning system. The other options investigated prevent customisation removing the ability

to adapt and transform the system to the analysis needs. The prototyping in chapter five, provides a reasonable solution and workable base to begin analysis. The application can parse a social media stream or textual web content and provide a valuable output in the form of categorisation and grouping, the demonstration of grouping and network discovery from PubMed articles demonstrate the flexibility of this approach. A key take away from this chapter is the system adaptability and simplicity to create meaningful outcomes. The discovery of potential user interface and user interaction solutions have highlighted the wide range of options that will allow a quality user interface to be created on an open-sourced platform. The chapter made an important discovery of training data and its role in the influence of quality outcomes, with further research required to determine the ability of a supervised system when compared to an unsupervised system.

7.3 Limitations with the research

The prototype that has been developed enables discovery of topics, trends and sentiment across textual inputs. However, the continual growth of the open-source machine learning field new python implementations makes large scale research difficult to maintain pace with the industry, as evidenced in the rapid rise of PyTorch, and Keras both highly capable of performing the role and with simpler implementations. The research demonstrates that algorithms and their python implementations are capable of performing machine learning categorisation on textual sources. The growth in computation power allows for real-time analysis. The advancement of machine learning during this thesis has further strengthened the notion that machine learning is now at a stage that more applications should be making use of machine learning. The growth of PyTorch and Keras will allow a more efficient and more accurate solution. Enabling a system which actively learns positive actions and outcomes whilst discovering actionable information to facilitate the decision-making process to enhance the patient's journey, and potentially optimise the overall treatment as an unsupervised system.

The prototype has been successful in proving the viability of making machine

learning accessible. The various small prototype experiments that have been created prove each concept can be adapted and simplified to perform a single operation. Each operation allows smaller tasks to be chained and create a much greater analysis tool. The manual update and training of the model prevents the systems self learning abilities and its ability to create groups or categories based upon that input, this manual supervised system is an issue that could be addressed by redevelopment under a new framework released.

7.4 FURTHER RESEARCH

The academic papers and prototypes produced have proven to the hospital that further development and a move towards a live trial is appropriate, as such a larger team have been composed and are to begin a new development phase based upon this research [93]. The new social discovery project utilises machine learning, to monitor social media text in combination with event information such as football fixtures or local city centre attractions or events that may cause a large number of visitors. Using this information the new project will attempt to predict A&E admissions along with possible admission reasons, creating valuable information as to the volume and type of admission that may be present due to an event or action.

7.4.1 PyTorch

The PyTorch framework has taken powerful machine learning algorithms and enabled a pure python implementation that allows customisation and creates simplified options for directing inputs and creating graphs based upon the outputs of the system. Flair [6]: A PyTorch-based framework for NLP tasks such as sequence tagging and classification. The core idea of the framework is to present a simple, unified interface for conceptually very different types of word and document embeddings. This effectively hides all embedding-specific engineering complexity and allows researchers to "mix and match" various embeddings with little effort. The framework also implements modern training and hyperparameter rou-

tines. The next version of the prototype is to use a full PyTorch implementation. Initial investigation of PyTorch reveals simplification of the building process as it removes a lot of dependencies required to build the current solution, the reduced python libraries will reduce the disk space requirements of the current solution. Pytorch allows a simple method for activation function change, the low amount of code can promote a pass down effect for end-users that will be given more flexibility on performance or speed. The next prototype will be changed from a sigmoid activation to ReLu and SoftMax for output of the neural network.

References

[1] Scikit-learn 2.2. manifold learning — scikit-learn 0.18.1documentation[online]. URL learn.org/stable/modules/manifold.html.

[2] Using patient feedback: A practical guide to improving patient experience. Technical report, Picker Institute, March 2009. URL http://www.nhssurveys.org/Filestore/documents/QIFull.pdf.

[3] Through big data: Using patient feedback: A practical guide to improving patient experience, March 2009.

[4] *MLWave: kepler-mapper: KeplerMapper is a Python Class for Visualization of High-Dimensional Data and 3-D Point Cloud Data.* . kepler-mapper, 2017.

[5] Saleem Abuleil and Khalid Alsamara. Using NLP approach for analyzing customer reviews. In *Computer Science & Information Technology (CS & IT)*, pages 117–124. Academy & Industry Research Collaboration Center (AIRCC), jan 2017. doi: 10.5121/csit.2017.70112.

[6] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4010. URL https://www.aclweb.org/anthology/N19-4010.

[7] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, January 2002. doi: 10.1103/revmodphys.74.47.

[8] Alteryx. Website. URL https://www.alteryx.com/.

[9] Anaconda. Website. URL https://www.anaconda.com/what-is-anaconda/.

[10] Oscar Araque, Ignacio Corcuera-Platas, J. Fernando Sánchez-Rada, and Carlos A. Iglesias. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications*, 77:236–246, jul 2017. doi: 10.1016/j.eswa.2017.02.002.

[11] Ayasdi, 2017. URL https://www.ayasdi.com.

[12] Antonia Azzini and Paolo Ceravolo. Consistent process mining over big data triple stores. In *2013 IEEE International Congress on Big Data*, volume 13, pages 54–61. IEEE, June 2013. doi: 10.1109/bigdata.congress.2013.17.

[13] Pierre Baldi. *Bioinformatics - The Machine Learning Approach 2e*. MIT Press, Cambridge, MA, 2001. ISBN 026202506X. URL https://www.ebook.de/de/product/3782781/pierre_baldi_bioinformatics_the_machine_learning_approach_2e.html.

[14] Michele Banko and Eric Brill. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics - ACL '01*, pages 26–33, Stroudsburg, PA, USA, February 2001. Association for Computational Linguistics. doi: 10.3115/1073012.1073017.

[15] Srividya K. Bansal and Sebastian Kagemann. Integrating big data: A semantic extract-transform-load framework. *Computer*, 48(3):42–50, March 2015. doi: 10.1109/mc.2015.76.

[16] S. Barabási A and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[17] David W. Bates, R. Scott Evans, Harvey Murff, Peter D. Stetson, Lisa Pizziferri, and George Hripcsak. Detecting adverse events using information technology. *Journal of the American Medical Informatics Association*, 10(2):115–128, mar 2003. doi: 10.1197/jamia.m1074.

[18] John Baylis and Klaus Janich. Topology. *The Mathematical Gazette*, 69 (448):149, June 1985. doi: 10.2307/3616955.

[19] Irad Ben-Gal. *Bayesian Networks*. American Cancer Society, 2008. ISBN 9780470061572. doi: 10.1002/9780470061572.eqr089. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470061572.eqr089.

[20] Mark J. Berger. Large scale multi-label text classification with semantic word vectors. 2015.

[21] Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. Moa: Massive online analysis. *J. Mach. Learn. Res.*, 11:1601–1604, August 2010. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1756006.1859903.

[22] S. Bird, E. Loper, and E. Klein. *Natural Language Processing with Python*. O'Reilly Media Inc, 2009.

[23] E. Blanco, N. Castell, and D. Moldovan. Causal relation extraction. In *proceedings of the Sixth International Conference on Language Resources andEvaluation (LREC 08)*, 2008.

[24] D. M. Blei, Y. Ng A., M. Jordan, and J. Lafferty. Latent dirichlet allocation. In *Journal of Machine Learning Research3*, volume 4, pages 993–1022. 2003.

[25] E H Bradley. Data feedback efforts in quality improvement: lessons learned from US hospitals. *Quality and Safety in Health Care*, 13(1):26–31, feb 2004. doi: 10.1136/qhc.13.1.26.

[26] Capgemini. The deciding factor: Big data & decision making. capgemini reports, 2012.

[27] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, jan 2009. doi: 10.1090/s0273-0979-09-01249-x.

[28] I. Carol and S. Britto Ramesh Kumar. Conflict identification and resolution in heterogeneous datasets: A comprehensive survey. *International Journal of Computer Applications*, 113(12):22–27, March 2015. doi: 10.5120/19879-1885.

[29] Y. Y. Chow. *Application of Data Analytics to Cyber Forensic Data A Major Qualifying Project Report*. MITRE Corporation, 2016.

[30] Jiang J. J. Conrath D W. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference Research on Computational Linguistics (ROCLING X)*, Taiwan, 1997.

[31] M. Cottle, S. Kanwal, M. Kohn, T. Strine, and N. Treister. Transforming health care through big data: Strategies for leveraging big data in the health care industry. report,[online]. available from. *Online*, 27:2017, 2013. URL bc6183f1c18e748a49b87a25911a0555.ssl.cf2.rackcdn.com/iHT2_BigData_2013.pdf.

[32] Marc Coudriau, Abdelkader Lahmadi, and Jerome Francois. Topological analysis and visualisation of network monitoring data: Darknet case study. In *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, December 2016. doi: 10.1109/wifs.2016.7823920.

[33] D. Danks and Griffiths T. L. Tenenbaum J B. *Dynamical Causal Learning*. NIPSMIT Press, 2002.

[34] Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy, May 2006. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2006/pdf/440_pdf.pdf.

[35] C. Dobre and Bessis N. . *Big Data and Internet of Things: A Roadmap for Smart Environments (Studies in Computational Intelligence Book 546)*. Springer, 2014. ISBN 978-3-319-05029-4. URL https://www.amazon.com/Big-Data-Internet-Things-Computational-ebook/dp/BooRZIH6J8?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=BooRZIH6J8.

[36] Xin Luna Dong and Felix Naumann. Data fusion. *Proceedings of the VLDB Endowment*, 2(2):1654–1655, August 2009. doi: 10.14778/1687553.1687620.

[37] M. Draper. Seeking consumer views: what use are results of hospital patient satisfaction surveys? *International Journal for Quality in Health Care*, 13(6): 463–468, December 2001. doi: 10.1093/intqhc/13.6.463.

[38] Clare Dyer. Alfie evans case: Proposed law aims to prevent conflicts between parents and doctors. *BMJ*, page k1895, apr 2018. doi: 10.1136/bmj.k1895.

[39] Edelsbrunner, Letscher, and Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, November 2002. doi: 10.1007/s00454-002-2885-2.

[40] H. Edelsbrunner and J. Harer. Computational topology: an introduction. *American Mathematical Soc*, 2010.

[41] Stephanie Evergreen and Chris Metzner. Design principles for data visualization in evaluation. *New Directions for Evaluation*, 2013(140):5–20, dec 2013. doi: 10.1002/ev.20071. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/ev.20071.

[42] M. Fayyad U., G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. Advances in knowledge discovery and data mining. *American Association for Artificial Intelligence*, 1996.

[43] Ronen Feldman and James Sanger. *The Text Mining Handbook*. Cambridge University Press, 2006. doi: 10.1017/cbo9780511546914.

[44] W.N. Francis, , and H. Kucera, editors. *The Brown Corpus: A Standard Corpus of Present-Day Edited American English*. Department of Linguistics, Brown University, Providence, RI, 1979.

[45] C Friedman and G Hripcsak. Natural language processing and its future in medicine. *Academic Medicine*, 74(8):890–5, aug 1999. doi: 10.1097/00001888-199908000-00012.

[46] Rie Gaku and Soemon Takakuwa. Big data-driven service level analysis for a retail store. In *Proceedings of the 2015 Winter Simulation Conference*, WSC '15, pages 791–799, Piscataway, NJ, USA, 2015. IEEE Press. ISBN 978-1-4673-9741-4. URL http://dl.acm.org/citation.cfm?id=2888619.2888709.

[47] Tejal K. Gandhi, Erin Graydon-Baker, Camilla Neppl Huber, Anthony D. Whittemore, and Michael Gustafson. Closing the loop: Follow-up and feedback in a patient safety program. *The Joint Commission Journal on Quality and Patient Safety*, 31(11):614–621, November 2005. doi: 10.1016/s1553-7250(05)31079-8.

[48] Joachim Giesen, Frederic Cazals, Mark Pauly, and Afra Zomorodian. The conformal alpha shape filtration. *The Visual Computer*, 22(8):531–540, jun 2006. doi: 10.1007/s00371-006-0027-1.

[49] J. E. Goodman. *Surveys on Discrete and Computational Geometry*. American Mathematical Society, 2008. ISBN 0821842390. URL https://www.ebook.de/de/product/35019116/surveys_on_discrete_and_computational_geometry.html.

[50] Felix Greaves, Daniel Ramirez-Cano, Christopher Millett, Ara Darzi, and Liam Donaldson. Machine learning and sentiment analysis of unstructured free-text information about patient experience online. *The Lancet*, 380:S10, November 2012. doi: 10.1016/s0140-6736(13)60366-9.

[51] Rajeev Gupta, Himanshu Gupta, and Mukesh Mohania. Cloud computing and big data analytics: What is new from databases perspective? In *Big Data Analytics*, pages 42–61. Springer Berlin Heidelberg, Heidelberg, 2012. doi: 10.1007/978-3-642-35542-4_5.

[52] A. Hagberg, D. Schult, and P. Swart. Exploring network structure, dynamics, and function using networkx. In Travis Vaught, , and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference (SciPy2008), Gael Varoquaux*, Pasadena, CA USA, 2008.

[53] R. Harrison, R. Lawton, and K. Stewart. Doctors' experiences of adverse events in secondary care: the professional and personal impact. *Clinical Medicine*, 14(6):585–590, dec 2014. doi: 10.7861/clinmedicine.14-6-585.

[54] Reema Harrison, Merrilyn Walton, Judith Healy, Jennifer Smith-Merry, and Coletta Hobbs. Patient complaints about hospital services: applying a complaint taxonomy to analyse and respond to complaints: Table 1. *International Journal for Quality in Health Care*, 28(2):240–245, January 2016. doi: 10.1093/intqhc/mzw003.

[55] Titov I. Henderson J. Incremental bayesian networks for structure prediction. URL http://cui.unige.ch/titov/papers/icml07.pdf.

[56] G. Holmes, A. Donkin, and I. H. Witten. Weka: a machine learning workbench. In *Proceedings of ANZIIS '94 - Australian New Zealnd Intelligent Information Systems Conference*, pages 357–361, Nov 1994. doi: 10.1109/ANZIIS.1994.396988.

[57] C.J. Hutto and Eric Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. 01 2015.

[58] I.B.M. Website. URL https://www.ibm.com/analytics/.

[59] Krzysztof Janowicz. Extending semantic similarity measurement with thematic roles. In *GeoSpatial Semantics*, volume 3799, pages 137–152. Springer Berlin Heidelberg, 2005. doi: 10.1007/11586180_10.

[60] Finn V. Jensen. Bayesian networks. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(3):307–315, nov 2009. doi: 10.1002/wics.48.

[61] H. S. Kaplan. Organization of event reporting data for sense making and system improvement. *Quality and Safety in Health Care*, 12 (90002):68ii–72, December 2003. ISSN 1475-3898. doi: 10.1136/qhc.12.suppl_2.ii68. URL https://qualitysafety.bmj.com/content/12/suppl_2/ii68.

[62] Adam Kilgarriff and Christiane Fellbaum. WordNet: An electronic lexical database. *Language*, 76(3):706, September 2000. doi: 10.2307/417141.

[63] Christine E. Kistler, Louise C. Walter, C. Madeline Mitchell, and Philip D. Sloane. Patient perceptions of mistakes in ambulatory care. *Archives of Internal Medicine*, 170(16), sep 2010. doi: 10.1001/archinternmed.2010.288.

[64] Aman Kumar, Hassan Alam, Rahul Kumar, and Shweta Sheel. Understanding medical named entity extraction in clinical notes. 2015.

[65] M. Lan, C. L. Tan, J. Su, and Y. Lu. Supervised and traditional term weighting methods for automatic text categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):721–735, April 2009. doi: 10.1109/TPAMI.2008.110.

[66] Steve Lavalle, Eric Lesser, Rebecca Shockley, Michael S. Hopkins, and Nina Kruschwitz. Big data, analytics and the path from insights to value. *MIT Sloan Management Review*, 52:21–32, 12 2011.

[67] S. LaVela and A. Gallan. Evaluation and measurement of patient experience. *Patient Experience Journal*, 1:28–36, 2014.

[68] Min Kyung Lee. Understanding perception of algorithmic decisions: Fairness, trust, and emotion in response to algorithmic management. *Big Data & Society*, 5(1):205395171875668, mar 2018. doi: 10.1177/2053951718756684.

[69] Carolyn Lees. Measuring the patient experience. *Nurse Researcher*, 19(1): 25–28, October 2011. doi: 10.7748/nr2011.10.19.1.25.c8768.

[70] Gondy Leroy, Trudi Miller, Graciela Rosemblat, and Allen Browne. A balanced approach to health information evaluation: A vocabulary-based naïve bayes classifier and readability formulas. *Journal of the American Society for Information Science and Technology*, 59(9):1409–1419, 2008. doi: 10.1002/asi.20837.

[71] Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2012.

[72] Bing Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, May 2012. doi: 10.2200/s00416ed1v01y201204hlt016.

[73] N. R. Llanwarne, G. A. Abel, M. N. Elliott, C. A. M. Paddison, G. Lyratzopoulos, J. L. Campbell, and M. Roland. Relationship between clinical quality and patient experience: Analysis of data from the english quality and outcomes framework and the national GP patient survey. *The Annals of Family Medicine*, 11(5):467–472, September 2013. doi: 10.1370/afm.1514.

[74] Jonathan Lomas. Using research to inform healthcare managers' and policy makers' questions: From summative to interpretive synthesis. *Healthcare Policy | Politiques de Santé*, 1(1):55–71, September 2005. doi: 10.12927/hcpol..17567.

[75] C. D. Manning. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[76] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014. URL http://www.aclweb.org/anthology/P/P14/P14-5010.

[77] Inocencio Daniel Maramba, Antoinette Davey, Marc N Elliott, Martin Roberts, Martin Roland, Finlay Brown, Jenni Burt, Olga Boiko, and John Campbell. Web-based textual analysis of free-text patient experience comments from a survey in primary care. *JMIR Medical Informatics*, 3(2):e20, may 2015. doi: 10.2196/medinform.3783.

[78] Microsoft. Website. URL https://azure.microsoft.com/en-gb/services/machine-learning-studio/.

[79] E. Molnar, N. Kryvinska, and M. Gregus. Customer driven big-data analytics for the companies' servitization. the spring servitization conference 2014. Technical report, May 2014.

[80] D. Morozov. Welcome to dionysus documentation![online]. available. URL http://www.mrzv.org/software/dionysus/.

[81] Harvey J Murff, Vimla L Patel, George Hripcsak, and David W Bates. Detecting adverse events for patient safety research: a review of current methodologies. *Journal of Biomedical Informatics*, 36(1-2):131–143, feb 2003. doi: 10.1016/j.jbi.2003.08.003.

[82] Daniel Müllner and Aravindakshan Babu. Python mapper: An open-source toolchain for data exploration, analysis and visualization, 2013. URL http://danifold.net/mapper.

[83] Prakash M. Nadkarni, Lucila Ohno-Machado, and Wendy W. Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, September 2011. doi: 10.1136/amiajnl-2011-000464.

[84] Srini Narayanan and Daniel Jurafsky. Bayesian models of human sentence processing. In *PROCEDINGS OF 20 TH ANNUAL CONFERENCE OF THE COGNITIVE SCIENCE SOCIETY*, pages 752–757. Erlbaum, 1998.

[85] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, January 2003. doi: 10.1137/s003614450342480.

[86] D. Niedermayer. An introduction to bayesian networks and their contemporary applications. URL http://www.niedermayer.ca/papers/bayesian/bayes.html. website.

[87] Philip Ogren and Steven Bethard. Building test suites for UIMA components. In *Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing (SETQA-NLP 2009)*, pages 1–4, Boulder, Colorado, June 2009. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W09-1501.

[88] H. A. Peacock. Family and friends tests may not give us the answers we're looking for. *BMJ*, 346(jun04 2):f3551–f3551, June 2013. doi: 10.1136/bmj.f3551.

[89] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Elsevier, 1988. doi: 10.1016/c2009-0-27609-4.

[90] Ted Pedersen. Integrating natural language subtasks with bayesian belief networks, 1999.

[91] Alexandre Pinto, Hugo Gonçalo Oliveira, and Ana Oliveira Alves. Comparing the Performance of Different NLP Toolkits in Formal and Social Media Text. In Marjan Mernik, José Paulo Leal, and Hugo Gonçalo Oliveira, editors, *5th Symposium on Languages, Applications and Technologies (SLATE'16)*, volume 51 of *OpenAccess Series in Informatics (OASIcs)*, pages 3:1–3:16, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-006-4. doi: 10.4230/OASIcs.SLATE.2016.3. URL http://drops.dagstuhl.de/opus/volltexte/2016/6008.

[92] F. PONSIGNON, A. SMART, M. WILLIAMS, HALL, and J. experience quality: an empirical exploration using content analysis techniques. *Journal of Service Management; Bingley*, 26(3):460–485, 2015.

[93] EHU Press office. Edge hill academic working to cut a&e waiting times. Online, September 2019. URL https://www.edgehill.ac.uk/news/2019/09/edge-hill-academic-working-to-cut-ae-waiting-times/.

[94] Wullianallur Raghupathi and Viju Raghupathi. Big data analytics in healthcare: promise and potential. *Health Information Science and Systems*, 2(1):3, February 2014. doi: 10.1186/2047-2501-2-3.

[95] Jeffrey Ray and Marcello Trovati. A survey of topological data analysis (TDA) methods implemented in python. In *Advances in Intelligent Networking and Collaborative Systems*, pages 594–600. Springer International Publishing, August 2017. doi: 10.1007/978-3-319-65636-6_54.

[96] Jeffrey Ray and Marcello Trovati. On the need for a novel intelligent big data platform: A proposed solution. In *Advances in Intelligent Networking and Collaborative Systems*, pages 473–478. Springer International Publishing, aug 2018. doi: 10.1007/978-3-319-98557-2_43.

[97] Jeffrey Ray, Marcello Trovati, and Simon Minford. A preliminary automated approach to assess hospital patient feedback. In *Advances in Intelligent Networking and Collaborative Systems*, pages 585–593. Springer International Publishing, aug 2017. doi: 10.1007/978-3-319-65636-6_53.

[98] Jeffrey Ray, Olayinka Johnny, Marcello Trovati, Stelios Sotiriadis, and Nik Bessis. The rise of big data science: A survey of techniques, methods and approaches in the field of natural language processing and network theory. *Big Data and Cognitive Computing*, 2(3):22, aug 2018. doi: 10.3390/bdcc2030022.

[99] R. Rehurek. Efficient topic modelling in python. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, ELRA*, pages 45–50, Valletta, Malta, 2010.

[100] Filipe N. Ribeiro, Matheus Araújo, Pollyanna Gonçalves, Marcos André Gonçalves, and Fabrício Benevenuto. Sentibench - a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, 5(1):23, Jul 2016. ISSN 2193-1127. doi: 10.1140/epjds/s13688-016-0085-1. URL https://doi.org/10.1140/epjds/s13688-016-0085-1.

[101] T. Richards. Listen to patients first. *BMJ*, 349(sep23 1):g5765–g5765, September 2014. doi: 10.1136/bmj.g5765.

[102] Matteo Rucco, Lorenzo Falsetti, Damir Herman, Tanya Petrossian, Emanuela Merelli, Cinzia Nitti, and Aldo Salvi. Using topological data analysis for diagnosis pulmonary embolism. September 2014.

[103] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. Earthquake shakes twitter users. In *Proceedings of the 19th international conference on World wide*

web - WWW '10, WWW '10, pages 851–860, New York, NY, USA, 2010. ACM Press. ISBN 978-1-60558-799-8. doi: 10.1145/1772690.1772777. URL http://doi.acm.org/10.1145/1772690.1772777.

[104] Olivia Sanchez-Graillet and Massimo Poesio. Acquiring Bayesian networks from text. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2004/pdf/240.pdf.

[105] Alexandra Schofield, Måns Magnusson, and David Mimno. Pulling out the stops: Rethinking stopword removal for topic models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, 2017. doi: 10.18653/v1/e17-2069.

[106] S D Scott, L E Hirschinger, K R Cox, M McCoig, J Brandt, and L W Hall. The natural history of recovery for the healthcare provider "second victim" after adverse patient events. *Quality and Safety in Health Care*, 18(5):325–330, oct 2009. doi: 10.1136/qshc.2009.032870.

[107] K. Senthilkumar and E. Ruchika Mehra Vijayan. Real time text analysis. *IOP Conference Series: Materials Science and Engineering*, 263:042005, nov 2017. doi: 10.1088/1757-899x/263/4/042005.

[108] Deborah Seys, Susan Scott, Albert Wu, Eva Van Gerven, Arthur Vleugels, Martin Euwema, Massimiliano Panella, James Conway, Walter Sermeus, and Kris Vanhaecht. Supporting involved health care professionals (second victims) following an adverse health event: A literature review. *International Journal of Nursing Studies*, 50(5):678–687, may 2013. doi: 10.1016/j.ijnurstu.2012.07.006.

[109] Alison Shepherd. Anger as alfie's army protests outside alder hey. *BMJ*, page k1801, apr 2018. doi: 10.1136/bmj.k1801.

[110] G. Singh, F. Memoli, and G Carlsson. Mapper: a topological mapping tool for point cloud data. In *in Eurographics symposium on point-based graphics*, 1991.

[111] R. Sirriyeh, R. Lawton, P. Gardner, and G. Armitage. Coping with medical error: a systematic review of papers to assess the effects of involvement in

medical errors on healthcare professionals' psychological well-being. *BMJ Quality & Safety*, 19(6):e43–e43, may 2010. doi: 10.1136/qshc.2009.035253.

[112] P. Spyns. Natural language processing in medicine: An overview. *Methods of Information in Medicine*, 35(04/05):285–301, sep 1996. doi: 10.1055/s-0038-1634681.

[113] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, June 2002. doi: 10.1162/106365602320169811.

[114] Mark Steyvers and Joshua B. Tenenbaum. The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth. *Cognitive Science*, 29(1):41–78, January 2005. doi: 10.1207/s15516709cog2901_3.

[115] Suricata, May 2017. URL https://Suricata-ids.org/.

[116] M. Trovati, J. Hayes, F. Palmieri, and N. Bessis. Automated extraction of fragments of bayesian networks from textual sources. *Applied Soft Computing*, 60:508–519, 2017.

[117] M. Trovati, E. Asimakopoulou, and N. Bessis. An investigation on human dynamics in enclosed spaces. *Journal of Computers and Electrical Engineering*, 2018.

[118] Marcello Trovati. Reduced topologically real-world networks. *International Journal of Distributed Systems and Technologies*, 6(2):13–27, April 2015. doi: 10.4018/ijdst.2015040102.

[119] Marcello Trovati and Ovidiu Bagdasar. Influence discovery in semantic networks: An initial approach. In *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*. IEEE, March 2014. doi: 10.1109/uksim.2014.48.

[120] Marcello Trovati and Nik Bessis. An influence assessment method based on co-occurrence for topologically reduced big data sets. *Soft Computing*, 20(5):2021–2030, February 2015. doi: 10.1007/s00500-015-1621-9.

[121] Marcello Trovati, Eleana Asimakopoulou, and Nik Bessis. An analytical tool to map big data to networks with reduced topologies. In *2014 International Conference on Intelligent Networking and Collaborative Systems*, pages 411–414. IEEE, September 2014. doi: 10.1109/incos.2014.25.

[122] Marcello Trovati, Nik Bessis, Anna Huber, Asta Zelenkauskaite, and Eleana Asimakopoulou. Extraction, identification, and ranking of network structures from data sets. In *2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems*, pages 331–337. IEEE, July 2014. doi: 10.1109/cisis.2014.46.

[123] Marcello Trovati, Aniello Castiglione, Nik Bessis, and Richard Hill. A kuramoto model based approach to extract and assess influence relations. In *Communications in Computer and Information Science*, pages 464–473. Springer Singapore, 2016. doi: 10.1007/978-981-10-0356-1_49.

[124] Edge Hill University. Research data management policy. Online, June 2018. URL https://www.edgehill.ac.uk/research/files/2018/06/RDM-Guidance-RO-GOV-14.pdf.

[125] Graham Upton and Ian Cook. *A Dictionary of Statistics*. Oxford University Press, January 2014. doi: 10.1093/acref/9780199679188.001.0001.

[126] Shari R. Veil, Tara Buehner, and Michael J. Palenchar. A work-in-process literature review: Incorporating social media in risk and crisis communication. *Journal of Contingencies and Crisis Management*, 19(2):110–122, apr 2011. doi: 10.1111/j.1468-5973.2011.00639.x.

[127] C. Lee Ventola. Social media and health care professionals: benefits, risks, and best practices. *P and T : a peer-reviewed journal for formulary management*, 39:491–520, July 2014. ISSN 1052-1372.

[128] Lidong Wang, Guanghui Wang, and Cheryl Ann Alexander. Big data and visualization: Methods, challenges and technology progress. *Digital Technologies*, 1(1):33–38, 2015. doi: 10.12691/dt-1-1-7. URL http://pubs.sciepub.com/dt/1/1/7.

[129] Xin Wang, Linpeng Huang, Xiaohui Xu, Yi Zhang, and Jun-Qing Chen. A solution for data inconsistency in data integration. *J. Inf. Sci. Eng.*, 27:681–695, 03 2011.

[130] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. In *The Structure and Dynamics of Networks*, volume 393, pages 440–442. Princeton University Press, December 2011. doi: 10.1515/9781400841356.301.

[131] Shmuel Weinberger. What is...persistent homology? *Notices of the American Mathematical Society*, 58, 01 2011.

[132] Albert W Wu and Rachel C Steckelberg. Medical error, incident investigation and the second victim: doing better but feeling worse? *BMJ Quality & Safety*, 21(4):267–270, jan 2012. doi: 10.1136/bmjqs-2011-000605.

[133] Yonghui Wu, Min Jiang, Jun Xu, Degui Zhi, and Hua Xu. Clinical named entity recognition using deep learning models. *AMIA ... Annual Symposium proceedings. AMIA Symposium*, 2017:1812–1819, 2017. ISSN 1942-597X.

[134] Wei Xie, Feida Zhu, Jing Jiang, Ee-Peng Lim, and Ke Wang. TopicSketch: Real-time bursty topic detection from twitter. *IEEE Transactions on Knowledge and Data Engineering*, 28(8):2216–2229, aug 2016. doi: 10.1109/tkde.2016.2556661.

[135] Wenpeng Yin, Katharina Kann, Mo Yu, and Hinrich Schütze. Comparative study of CNN and RNN for natural language processing. February 2017.

[136] Du Zhang. On temporal properties of knowledge base inconsistency. In *Transactions on Computational Science V*, volume 5540, pages 20–37. Springer Berlin Heidelberg, 2009. doi: 10.1007/978-3-642-02097-1_2.

[137] DU ZHANG. GRANULARITIES AND INCONSISTENCIES IN BIG DATA ANALYSIS. *International Journal of Software Engineering and Knowledge Engineering*, 23(06):887–893, aug 2013. doi: 10.1142/s0218194013500241.

[138] Shanshan Zhang and Slobodan Vucetic. Semi-supervised Discovery of Informative Tweets During the Emerging Disasters. *arXiv e-prints*, art. arXiv:1610.03750, Oct 2016.

Appendix

VADER & Twitter Live streaming code

```
import pandas as pd
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from nltk import tokenize
from nltk import pos_tag
import nltk
import re
import numpy as np
import tweepy
from tweepy import Stream
import pandas as pd
from tweepy.streaming import StreamListener
import unicodecsv as csv
import json
import matplotlib.pyplot as plt
import matplotlib.animation as animation
import time


#User account credentials in CSV file must 2 column wide
# - 2nd containing secret keys
#must be in order:
#consumer_key, consumer_secret, access_token, access_secret.
df = pd.read_csv('creds.csv')

consumer_key = df.iat[0, 1]
consumer_secret = df.iat[1, 1]
access_token = df.iat[2, 1]
access_secret = df.iat[3, 1]

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)

auth.set_access_token(access_token, access_secret)

api = tweepy.API(auth)
```

```
# api.wait -  to prevent 420 errors from twitter.
# will automatically rate for standard search -
# streaming is rate limited from Twitter.
api.wait_on_rate_limit = True
api.wait_on_rate_limit_notify = True


# public_tweets = api.home_timeline()
# for tweets in public_tweets:
#     print(tweets.text)

class listener(StreamListener):
    global rawTweet
    tweetcounter = 0  # Static Variable
    global dframe

    pos = 0.0
    neg = 0.0
    neu = 0.0

    getCompund = 0.0
    compoundCount = 1

    def on_data(self, data):
        all_data = json.loads(data)
        # collect all desired data fields
        if 'text' in all_data:
            tweet = all_data["text"]
            created_at = all_data["created_at"]
            retweeted = all_data["retweeted"]
            username = all_data["user"]["screen_name"]
            user_tz = all_data["user"]["time_zone"]
            user_location = all_data["user"]["location"]
            user_coordinates = all_data["coordinates"]

            #if coordinates are not present store blank
    value
            #otherwise get the coordinates.coordinates value
            if user_coordinates is None:
                final_coordinates = user_coordinates
            else:
                final_coordinates = str(all_data
                ["coordinates"]["coordinates"])

            # print((created_at, username, tweet))
            rawTweet = [{'Time': created_at},
```

```
            {'User': username}, {'Tweet': tweet}, {
                'Location': user_location},
                {'Coordinates': user_coordinates}]
            tokenData = tokenize.sent_tokenize(tweet)
            wordtoken = nltk.word_tokenize(tweet)


            # Sent data
            sid = SentimentIntensityAnalyzer()
            # Print Raw tweet
            print(tweet)
            # Polarity Score
            ss = sid.polarity_scores(tweet)
            pt = pos_tag(tokenData)
            postweet = pos_tag(wordtoken)

            # print(pt)
            print(postweet)
            print(ss)
            for k in sorted(ss):
                print('{0}: {1}, '.format(k, ss[k]), end='')

            # counting pos neg and neut
            if ss.get('compound') > 0.0:
                listener.pos += 1
                print('pos = {}'.format(listener.pos))
            if ss.get('compound') == 0.0:
                listener.neu += 1
                print('neu = {}'.format(listener.neu))
            if ss.get('compound') < 0.0:
                listener.neg += 1
                print('neg = {}'.format(listener.neg))

            # print(rawTweet)
            with open('count.txt', 'a') as countfile:
                countfile.truncate()
                countfile.write('{}, {}'.format(
                    listener.compoundCount,
                    ss.get('compound')))
                listener.compoundCount += 1
                countfile.write('\n')
                countfile.flush()
            with open('totals.txt', 'a') as totalfile:
                totalfile.write('{},{},{}'.format(
                    listener.pos, listener.neu, listener.neg
))
```

```
125                 totalfile.flush()
126
127             with open('output.json', 'a') as outfile:
128                 json.dump((rawTweet), outfile)  #
129                 outfile.write('\n')
130                 json.dump((ss), outfile)
131                 outfile.write('\n')
132                 outfile.close()
133
134             # Counter
135             listener.tweetcounter += 1
136
137             if listener.tweetcounter < listener.stopat:
138
139                 return True
140             else:
141                 print
142             ('Totals (Pos: {})(Nut: {})(Neg: {})(Tweets: {})
    '
143             .format(
144                     listener.pos,
145                     listener.neu,
146                     listener.neg,
147                     listener.pos
148                     + listener.neu
149                     + listener.neg))
150                 open('count.txt', 'w').close()
151                 open('totals.txt', 'w').close()
152                 with open('output.json', 'a') as outfile:
153                     outfile.write
154                 ('Totals: Pos: {}, Neut: {}, Neg:{}'.format(
155                         listener.pos,
156                         listener.neu,
157                         listener.neg))
158                     outfile.close()
159                 return False
160
161             # DF for graph
162
163         else:
164             return True
165
166     def on_status(self, status):
167         # Prints the text of the tweet
168         newTweet = ('Author: ({}) Tweet: {} Time: {}'.format
    (
```

```
            status.author.screen_name,
            status.text,
            status.created_at))

        # print(newTweet)

        # writesvto .csv file defined below,
        # can add other status points e.g. time_stamp.

        # a counter to limit the amount of collected tweets
        # limit set @ listener.stopat

        listener.tweetcounter += 1
        if listener.tweetcounter < listener.stopat:

            return True
        else:

            return False

            # There are many options in the status object,
            # hashtags can be very easily accessed.
            # for hashtag in status.entries['hashtags']:
            # print(hashtag['text'])

    # Error received from twitter API call
    def on_error(self, status_code):
        print('Got an error with status code: '
        + str(status_code))
        return True  # To continue listening

    # Time out from twiter API
    def on_timeout(self):
        print('Timeout...')
        return True  # To continue listening


# set Tweet limit and terms to be searched
# listener.stopat = max tweets to be collected.
# twitterStream.filter = search terms to be included
# in ' ' and , separated.

def getTweets():
    try:

        listener.stopat = 30
```

```
        twitterStream = Stream(auth, listener())
        twitterStream.filter(track=['Trump'],
        async=True, languages=['en'])

    except KeyboardInterrupt:
        print('KB INTERRUPTED')


# start process_tweet
def processTweet(tweet):
    # process the tweets

    # Convert to lower case
    tweet = tweet.lower()
    # Convert www.* or https?://* to URL
    tweet = re.sub('((www\.[^\s]+)|(https?://[^\s]+))',
    'URL',
    tweet)
    # Convert @username to AT_USER
    tweet = re.sub('@[^\s]+', 'AT_USER', tweet)
    # Remove additional white spaces
    tweet = re.sub('[\s]+', ' ', tweet)
    # Replace #word with word
    tweet = re.sub(r'#([^\s]+)', r'\1', tweet)
    # trim
    tweet = tweet.strip('\'"')
    return tweet


getTweets()


fig = plt.figure()
ax1 = fig.add_subplot(1, 1, 1)


def animate(i):

    pullData = open("count.txt", "r").read()
    dataArray = pullData.split('\n')
    xar = []
    yar = []
    for eachLine in dataArray:
        if len(eachLine) > 1:
            x, y = eachLine.split(',')
            xar.append(float(x))
```

```
            yar.append(float(y))
            ax1.clear()
            ax1.plot(xar, yar)
            # plt.ylabel('Polarity')
            # plt.xlabel('Tweet Number')


ani = animation.FuncAnimation(fig, animate, interval=1000)
plt.show()
```

Listing 7.1: VADER sentiment

Neural Network & NLTK code

```
import nltk
from nltk.stem.lancaster import LancasterStemmer
import os
import json
import datetime
stemmer = LancasterStemmer()
import numpy as np
import time

training_data = []
#Training sets - Alderhey_events
training_data.append({"class":"AlderHey_event", "sentence":"
    Thanks so much @AlderHey for presenting with us at #
    MUN17 to showcase our Interoperability solutions!"})
training_data.append({"class":"AlderHey_event", "sentence":"
    The big charity fundraiser event takes place tomorrow
    @AlderHey Hospital"})
training_data.append({"class":"AlderHey_event", "sentence":"
    9-13th October. We'll be joining in a sponsored read for
     Alder Hey Childrens' Hospital- a place very close to
    our hearts."})
training_data.append({"class":"AlderHey_event", "sentence":"
    We have had an amazing night fundraising for @AlderHey
    thank you to everyone for your support-you are all
    amazing"})
# Thank you
training_data.append({"class":"AlderHey_thank", "sentence":"
    Today is #worldheartday2017. Thank you @AlderHey for
    making my nephew the bravest little boy I have ever
    known. My little superhero "})
training_data.append({"class":"AlderHey_thank", "sentence":"
    A good day to thank @AlderHey saving lives in poorer
    countries."})
```

```
training_data.append({"class":"AlderHey_thank", "sentence":"
    huge thank you for their donation your donation for
    @AlderHey means a lot"})
training_data.append({"class":"AlderHey_thank", "sentence":"
    Big thank you to my three helpers Paula, Alex and Sam-
    we loved the circus and collected for Alder Hey at the
    end."})
# political
training_data.append({"class":"Other","sentence":"More
    trouble brewing over Alder Hey housing plans"})
training_data.append({"class":"Other","sentence":"Alder Hey
    will bring back housing plans -residents vow 2 fight NO
    MORE DONATIONS ALDERHEY WE WANT R PARK  NOT HOMES'"})
training_data.append({"class":"Other","sentence":"@AlderHey
    need to get grip on smokers outside hospital. I'm the
    one having to take my trachy son off grounds to have a
    coffee n play in sun "})

#staff
training_data.append({"class":"staff","sentence":"I'm
    utterly convinced the speedy diagnosis at Alder Hey was
    responsible for full recovery. #encephalitis"})
training_data.append({"class":"staff","sentence":"The Staff
    have been amazing today"})
training_data.append({"class":"staff","sentence":"The great
    work you do @AlderHey for children and their families is
     absolutely amazing! God Bless"})
training_data.append({"class":"staff","sentence":"Joke that
    @AlderHey has 1 part time knee specialist.3months to get
     appointment,which is another 3months away,then 4months
    if op required"})

words = []
classes = []
documents = []
ignore_words = ['?',',','.']
# loop through each sentence in our training data
for pattern in training_data:
    # tokenize each word in the sentence
    w = nltk.word_tokenize(pattern['sentence'])
    # add to our words list
    words.extend(w)
    # add to documents in our corpus
    documents.append((w, pattern['class']))
    # add to our classes list
    if pattern['class'] not in classes:
```

```
        classes.append(pattern['class'])

# stem and lower each word and remove duplicates
words = [stemmer.stem(w.lower()) for w in words if w not in
    ignore_words]
words = list(set(words))

# remove duplicates
classes = list(set(classes))

#DEBUG FOR DOCS CLASS WORD
# print (len(documents), "documents")
# print (len(classes), "classes", classes)
# print (len(words), "unique stemmed words", words)


training = []
output = []
# create an empty array for our output
output_empty = [0] * len(classes)

# training set, bag of words for each sentence
for doc in documents:
    # initialize our bag of words
    bag = []
    # list of tokenized words for the pattern
    pattern_words = doc[0]
    # stem each word
    pattern_words = [stemmer.stem(word.lower()) for word in
    pattern_words]
    # create our bag of words array
    for w in words:
        bag.append(1) if w in pattern_words else bag.append
    (0)

    training.append(bag)
    # output is a '0' for each tag and '1' for current tag
    output_row = list(output_empty)
    output_row[classes.index(doc[1])] = 1
    output.append(output_row)

# sample training/output
i = 0
w = documents[i][0]
# DEBUG OUTPUT
# print ([stemmer.stem(word.lower()) for word in w])
```

```
# print (training[i])
# print (output[i])


# compute sigmoid nonlinearity
def sigmoid(x):
    output = 1 / (1 + np.exp(-x))
    return output


# convert output of sigmoid function to its derivative
def sigmoid_output_to_derivative(output):
    return output * (1 - output)


def clean_up_sentence(sentence):
    # tokenize the pattern
    sentence_words = nltk.word_tokenize(sentence)
    # stem each word
    sentence_words = [stemmer.stem(word.lower()) for word in
    sentence_words]
    return sentence_words


# return bag of words array: 0 or 1 for each word in the bag
    that exists in the sentence
def bow(sentence, words, show_details=False):
    # tokenize the pattern
    sentence_words = clean_up_sentence(sentence)
    # bag of words
    bag = [0] * len(words)
    for s in sentence_words:
        for i, w in enumerate(words):
            if w == s:
                bag[i] = 1
                if show_details:
                    print("found in bag: %s" % w)

    return (np.array(bag))


def think(sentence, show_details=False):
    x = bow(sentence.lower(), words, show_details)
    if show_details:
        print("sentence:", sentence, "\n bow:", x)
    # input layer is our bag of words
```

```
    l0 = x
    # matrix multiplication of input and hidden layer
    l1 = sigmoid(np.dot(l0, synapse_0))
    # output layer
    l2 = sigmoid(np.dot(l1, synapse_1))
    return l2


def train(X, y, hidden_neurons=10, alpha=1, epochs=50000, dropout=False, dropout_percent=0.5):
    print("Training with %s neurons, alpha:%s, dropout:%s %s" % (
    hidden_neurons, str(alpha), dropout, dropout_percent if dropout else ''))
    print("Input matrix: %sx%s    Output matrix: %sx%s" % (len(X), len(X[0]), 1, len(classes)))
    np.random.seed(1)

    last_mean_error = 1
    # randomly initialize our weights with mean 0
    synapse_0 = 2 * np.random.random((len(X[0]), hidden_neurons)) - 1
    synapse_1 = 2 * np.random.random((hidden_neurons, len(classes))) - 1

    prev_synapse_0_weight_update = np.zeros_like(synapse_0)
    prev_synapse_1_weight_update = np.zeros_like(synapse_1)

    synapse_0_direction_count = np.zeros_like(synapse_0)
    synapse_1_direction_count = np.zeros_like(synapse_1)

    for j in iter(range(epochs + 1)):

        # Feed forward through layers 0, 1, and 2
        layer_0 = X
        layer_1 = sigmoid(np.dot(layer_0, synapse_0))

        if (dropout):
            layer_1 *= np.random.binomial([np.ones((len(X), hidden_neurons))], 1 - dropout_percent)[0] * (
            1.0 / (1 - dropout_percent))

        layer_2 = sigmoid(np.dot(layer_1, synapse_1))

        # how much did we miss the target value?
        layer_2_error = y - layer_2
```

```
172
173         if (j % 10000) == 0 and j > 5000:
174             # if this 10k iteration's error is greater than
    the last iteration, break out
175             if np.mean(np.abs(layer_2_error)) <
    last_mean_error:
176                 print("delta after " + str(j) + " iterations
    :" + str(np.mean(np.abs(layer_2_error))))
177                 last_mean_error = np.mean(np.abs(
    layer_2_error))
178             else:
179                 print("break:", np.mean(np.abs(layer_2_error
    )), ">", last_mean_error)
180                 break
181
182         # in what direction is the target value?
183         # were we really sure? if so, don't change too much.
184         layer_2_delta = layer_2_error *
    sigmoid_output_to_derivative(layer_2)
185
186         # how much did each l1 value contribute to the l2
    error (according to the weights)?
187         layer_1_error = layer_2_delta.dot(synapse_1.T)
188
189         # in what direction is the target l1?
190         # were we really sure? if so, don't change too much.
191         layer_1_delta = layer_1_error *
    sigmoid_output_to_derivative(layer_1)
192
193         synapse_1_weight_update = (layer_1.T.dot(
    layer_2_delta))
194         synapse_0_weight_update = (layer_0.T.dot(
    layer_1_delta))
195
196         if (j > 0):
197             synapse_0_direction_count += np.abs(
198                 ((synapse_0_weight_update > 0) + 0) - ((
    prev_synapse_0_weight_update > 0) + 0))
199             synapse_1_direction_count += np.abs(
200                 ((synapse_1_weight_update > 0) + 0) - ((
    prev_synapse_1_weight_update > 0) + 0))
201
202         synapse_1 += alpha * synapse_1_weight_update
203         synapse_0 += alpha * synapse_0_weight_update
204
205         prev_synapse_0_weight_update =
```

```
    synapse_0_weight_update
206         prev_synapse_1_weight_update =
    synapse_1_weight_update
207
208     now = datetime.datetime.now()
209
210     # persist synapses
211     synapse = {'synapse0': synapse_0.tolist(), 'synapse1':
    synapse_1.tolist(),
212                'datetime': now.strftime("%Y-%m-%d %H:%M"),
213                'words': words,
214                'classes': classes
215                }
216     synapse_file = "synapses.json"
217
218     with open(synapse_file, 'w') as outfile:
219         json.dump(synapse, outfile, indent=4, sort_keys=True
    )
220     print("saved synapses to:", synapse_file)
221
222 #MODEL BUILD
223 X = np.array(training)
224 y = np.array(output)
225
226 start_time = time.time()
227
228 train(X, y, hidden_neurons=20, alpha=0.1, epochs=100000,
    dropout=False, dropout_percent=0.2)
229
230 elapsed_time = time.time() - start_time
231 print ("processing time:", elapsed_time, "seconds")
232
233 # probability threshold
234 ERROR_THRESHOLD = 0.2
235 # load our calculated synapse values
236 synapse_file = 'synapses.json'
237 with open(synapse_file) as data_file:
238     synapse = json.load(data_file)
239     synapse_0 = np.asarray(synapse['synapse0'])
240     synapse_1 = np.asarray(synapse['synapse1'])
241
242 def classify(sentence, show_details=False):
243     results = think(sentence, show_details)
244
245     results = [[i,r] for i,r in enumerate(results) if r>
    ERROR_THRESHOLD ]
```

```
    results.sort(key=lambda x: x[1], reverse=True)
    return_results =[[classes[r[0]],r[1]] for r in results]
    print ("%s \n classification: %s" % (sentence,
   return_results))
    return return_results

classify("The charity fundraising event is tomorrow at
   @Alderhey")
classify("My mate is doing a charity hike for alder hey, any
    donations greatly appreciated.")
classify("The nurse has been such a great help today")
print()
```

TSNE Word vector

```
import pandas as pd
pd.options.mode.chained_assignment = None
import numpy as np
import re
import nltk
from gensim.models import word2vec
import os
import sklearn
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt

import kmapper as km

data = pd.read_csv('nhs.csv', encoding='utf-8', delimiter='|
   ')
data['text'] = data['text'].astype(str)

#data = pd.read_table("/Users/jeffreyray/Dropbox/Python/t-
   sne-wordvector/1.txt",skip_blank_lines=True, encoding='
   latin-1', error_bad_lines=False, engine='python',
   delimiter='\n')
# data = pd.read_csv('csv_tweetsclass.csv', encoding='latin
   -1')

STOP_WORDS = nltk.corpus.stopwords.words()

def clean_sentence(val):
    regex = re.compile('([^\s\w]|_)+')
    sentence = regex.sub('', val).lower()
    sentence = sentence.split(" ")

    for word in list(sentence):
```

```
        if word in STOP_WORDS:
            sentence.remove(word)

    sentence = " ".join(sentence)
    return sentence

def clean_dataframe(data):
    data = data.dropna(how="any")

    # for col in ['Positive', 'Negative']:
    #     data[col] = data[col].apply(clean_sentence)
    # for col in ['Tweet']:
    #     data[col] = data[col].apply(clean_sentence)
    for col in ['text']:
        data['text'] = data['text'].apply(clean_sentence)
    return data
print('Cleaning Data: ')
#data = clean_dataframe(data)
print(data.head(5))

def build_corpus(data):
    corpus = []
    for col in ['text']:
    #for col in ['Positive', 'Negative']:
        for sentence in data['text'].iteritems():
            word_list = sentence[1].split(" ")
            corpus.append(word_list)
    # for col in ['Tweet']:
    #     for sentence in data[col].iteritems():
    #         word_list = sentence[1].split(" ")
    #         corpus.append(word_list)

    return corpus

corpus = build_corpus(data)
print('CORPUS: ')
print(corpus[0:2])

print('Building Model ...')
model = word2vec.Word2Vec(corpus, size=400, window=3,
    min_count=10, workers=5)
# print(model.wv['nothing'])

def tsne_plot(model):
    labels = []
    tokens = []
```

```
73
74      for word in model.wv.vocab:
75          tokens.append(model[word])
76          labels.append(word)
77
78      tsne_model = TSNE(perplexity=50, n_components=2, init='
        pca', n_iter=3500, random_state=42, learning_rate=500)
79      new_values = tsne_model.fit_transform(tokens)
80
81      x = []
82      y = []
83      for value in new_values:
84          x.append(value[0])
85          y.append(value[1])
86
87      plt.figure(figsize=(16, 16))
88      for i in range(len(x)):
89          plt.scatter(x[i], y[i])
90          plt.annotate(labels[i], xy=(x[i], y[i]), xytext= (5,
        2),
91                       textcoords='offset points', ha='right',
        va='bottom')
92      plt.show()
93
94  #print('Plotting Model: ')
95  #tsne_plot(model)
96
97  print(model.wv)
98  labels = []
99  tokens = []
100 def labtok(model):
101
102     for word in model.wv.vocab:
103         tokens.append(model[word])
104         labels.append(word)
105
106     return labels, tokens
107
108 labtok(model)
109
110 mapper = km.KeplerMapper(verbose=2)
111
112 projected_data = mapper.fit_transform(tokens, projection=
        sklearn.manifold.TSNE())
113
114 graph = mapper.map(projected_data, clusterer=sklearn.cluster
```

```
    .DBSCAN(eps=0.3, min_samples=5))

mapper.visualize(graph, path_html="trump.html",
    graph_gravity=0.25, custom_tooltips=np.array(labels))
print('Fin.')
```

LDA Mapping

```
import logging
import string
from time import time

import gensim
import pandas as pd
import pyLDAvis.gensim
from gensim import corpora
from gensim.parsing.preprocessing import split_alphanum
from nltk.corpus import stopwords
from nltk.stem.wordnet import WordNetLemmatizer

stop = set(stopwords.words('english'))
exclude = set(string.punctuation)
lemma = WordNetLemmatizer()

#f = open('/Users/jeffreyray/Dropbox/Python/topic modeling
    /1.txt')

#This is the input file mapped to a Pandas Data frame
df = pd.read_table("/Users/jeffreyray/Dropbox/Python/topic
    modeling/1.txt",skip_blank_lines=True, encoding='latin-1
    ', error_bad_lines=False, engine='python', delimiter='\n
    ')
#df = pd.read_csv('/Users/jeffreyray/Dropbox/Python/topic
    modeling/PN-all.csv', encoding='latin-1')
#This is the col from the data frame / csv file with terms
    to be analysed
posdoc = df['Introduction.']
#posdoc = df['Positive']
print(df)
def clean(doc):
    stop_free = " ".join([i for i in doc.lower().split() if
    i not in stop])
    punc_free = ''.join(ch for ch in stop_free if ch not in
    exclude)
    normalized = " ".join(lemma.lemmatize(word) for word in
    punc_free.split())
    normalized = split_alphanum(normalized)
```

```
    return normalized

#posdoc can be changed to match correct col line 17
doc_clean = [clean(doc).split() for doc in posdoc]

# text_pos = df.applymap(clean)['Positive']
# text_pos_list = [i.split() for i in text_pos]
# Logging
logging.basicConfig(format='%(asctime)s : %(levelname)s : %(message)s', level=logging.INFO,
                    filename='running.log', filemode='w')
# Creating the term dictionary of our courpus, where every unique term is assigned an index.
# dictionary = corpora.Dictionary(text_pos_list)
dictionary = corpora.Dictionary(doc_clean)
dictionary.save('posdictionary.dict')

# Converting list of documents (corpus) into Document Term Matrix using dictionary prepared above.
doc_term_matrix = [dictionary.doc2bow(doc) for doc in doc_clean]
corpora.MmCorpus.serialize('corpus.mm', doc_term_matrix)

# logging the start time
start = time()
# creating the object for LDA model using gensim
lda = gensim.models.LdaModel
# running and training lda model on document term matrix
ldamodel = lda(doc_term_matrix, num_topics=20, id2word=dictionary, passes=100)
print('used: {:.2f}s'.format(time() - start))
print(ldamodel.print_topics(num_topics=20, num_words=20))

for i in ldamodel.print_topics():
    for j in i: print(j)

# saving a trained model
ldamodel.save('topic.model')


# loading
# from gensim.models import LdaModel
# loading = ldamodel.load('topic.model')
# print(loading.print_topics(num_topics=2, num_words=4))

# new doc helper
```

```
def newdoc(doc):
    doc_clean = [clean(doc).split() for doc in posdoc]
    doc_term_matrix = [dictionary.doc2bow(doc) for doc in
    doc_clean]
    return doc_term_matrix


# pyLDAvis.enable_notebook()
d = gensim.corpora.Dictionary.load('posdictionary.dict')
c = gensim.corpora.MmCorpus('corpus.mm')
lda = gensim.models.LdaModel.load('topic.model')

data = pyLDAvis.gensim.prepare(lda, c, d)
data
pyLDAvis.show(data)
pyLDAvis.save_html(data, '/Users/jeffreyray/Dropbox/Python/
    topic modeling/newvis.html')
print('Finished')
```

Web Extraction

```
import nltk
from bs4 import BeautifulSoup, SoupStrainer
import requests




def get_pre_defined_webpages():
    web = ['http://www.fsb.org.uk', 'http://www.accaglobal.
    com', 'http://www.accaglobal.com', 'http://www.
    cimaglobal.com', 'http://www.fiscalsolutions.co.uk', '
    http://www.jackross.com', 'http://www.jacrox.co.uk', '
    http://www.slasscom.lk', 'http://www.
    bookkeepingaccounting.co.uk', 'http://www.skpgroup.com']

    return web

'''
The next methods extract text from an input webpage. The
    text is subsequently analysed to return
sequences of consecutive nouns. For example King Arthur will
     be considered as a single entity, rather than King and
    then
Arthur. No NER is carried out here, and this approach is
    rather naive. However, it seems to work fine for now.
```

```
'''
def get_text_from_webpages(input_WebPage):
    webPage = requests.get(input_WebPage)
    html = webPage.text

    soup = BeautifulSoup(html, 'html.parser')
    text = []
    output_text = []

    text = soup.find_all('p')
    for sentence in text:
      raw_text = sentence.get_text()
      if raw_text != '':
        output_text.append(raw_text)


    return output_text


def split_sentences(text):
    return text.split(".")


def text_analysis(sentence):
    # POS tagging
    tokens = nltk.word_tokenize(sentence)
    tagged = nltk.pos_tag(tokens)
    return tagged


def find_nouns(sentence):
    NNs = []
    try:
        analysis = text_analysis(sentence)
        for item in analysis:
            # if 'NN' in item[1] and len(item[0])>1:
                    # This only identifies nouns
            if 'NN' in item[1] or 'JJ' in item[1] and len(
    item[0]) > 1:  # This identifies nouns and adjectives
                NNs.append([item[0], analysis.index(item)])
    except:
        pass
    return NNs


def get_consecutive_items_final(input_list):
```

```
    # It identifies consecutive nouns in the input text
    output = []
    temp = []
    if input_list != []:
        temp.append(input_list[0][0])
        length = len(input_list) - 1
        index = 1
        while index <= length:
            if input_list[index][1] == input_list[index - 1][1] + 1:
                temp.append(input_list[index][0])
            else:
                output.append(temp)
                temp = []
                temp.append(input_list[index][0])
            index = index + 1
        output.append(temp)
    return output


def clean_nouns(noun_list):
    # It removes duplicates and words contained in concepts defined by more than one word
    temp = []
    temp_to_remove = []
    for noun in noun_list:
        if ' ' + noun in noun_list:
            noun_list.remove(' ' + noun)
        if noun + ' ' in noun_list:
            noun_list.remove(noun + ' ')
        if ' ' + noun + ' ' in noun_list:
            noun_list.remove(' ' + noun + ' ')
    for noun in noun_list:
        if noun not in temp:
            temp.append(noun)

    for index in range(len(temp)):
        for item in temp:
            if temp[index] in item and temp[index] != item:
                temp_to_remove.append(temp[index])
    for item in temp_to_remove:
        if item in temp:
            temp.remove(item)
    return temp


```

```
def get_nouns(text):
    noun_groups = []
    text_analysis_output = []
    for line in split_sentences(text):
        if line != '':
            text_analysis_output.append(find_nouns(line))
    for item in text_analysis_output:
        L = get_consecutive_items_final(item)
        for noun in L:
            if "<" not in noun and ">" not in noun and "br" not in noun and "]" not in noun and "[" not in noun:  # This doesn't filter out non-ASCII characters. I need to make sure we have the correct unicode settings
                if len(
                        noun) > 1:  # If this item contains two or more consecutive nouns, then group them together and add it to the noun_group list
                    temp = ''
                    for subnoun in noun:
                        temp = temp + ' ' + subnoun
                        noun_groups.append(temp)
                else:  # If this item contains only one noun then add it to the noun_group list
                    noun_groups.append(noun[0])

    return clean_nouns(noun_groups)


def nouns_from_set_of_webpages():
  output = []
  for web in get_pre_defined_webpages():
    temp = extract(web)
    if temp !=[]:
      output.append(temp)
  return output



'''
Main method.
'''

def extract(webpage):
  analysis = []
  text_from_webpage = get_text_from_webpages(webpage)
  for text_components in text_from_webpage:
```

```
    temp = get_nouns(split_sentences(text_components)[0])
    for item in temp:
      if item not in analysis and item !='':      #Remove empty string and duplication
        analysis.append(item)
  return analysis

def extract_nouns_from_webpages(webpage):
  output = []
  output.append(extract(webpage))
  # temp = nouns_from_set_of_webpages()
  # for item in temp:
  #   output.append(item)
  return output

# if __name__ == "__main__":

#     print extract('https://epsrc.ukri.org/funding/applicationprocess/routes/newac/nia/')

#     # Run the following if we have a set of websites to search.

#     print nouns_from_set_of_webpages()
```

User Interface

```
import pandas as pd
from collections import Counter
import logging
# from bokeh.transform import factor_cmap, linear_cmap
# from bokeh.models import ColumnDataSource, Range1d, LabelSet, Label
# from bokeh.models import HoverTool
from bokeh.plotting import figure, show, output_file, ColumnDataSource
from bokeh.palettes import inferno
import asyncio
import concurrent.futures
import extract_nouns_websites
import os
import sys
import getopt
import datetime
import codecs
import shutil
import json
```

```
import pprint
from googleapiclient.discovery import build
import tkinter as tk
from tkinter import ttk
from tkinter import scrolledtext
from tkinter import Menu
from tkinter import messagebox as mBox  # Message box / popup
import pandas as pd
import got3 as got
import datetime
import Classification

df = pd.read_csv('Keywords.csv')

logging.basicConfig()

# Decorator for threading and returning values
_DEFAULT_POOL = concurrent.futures.ThreadPoolExecutor()


def threadpool(f, executor=None):
    # @wraps(f)
    def wrap(*args, **kwargs):
        return asyncio.wrap_future((executor or _DEFAULT_POOL).submit(f, *args, **kwargs))

    return wrap


# Build main window & title
root = tk.Tk()  # --- USED FOR mBox display without extra window
root.withdraw()  # --- ^ ^ ^
win = tk.Tk()
win.title('Angels Data')

# Tab Control introduced here --------------------------------------
tabControl = ttk.Notebook(win)          # Create Tab Control

tab1 = ttk.Frame(tabControl)            # Create a tab
tabControl.add(tab1, text='Search')      # Add the tab

# tab2 = ttk.Frame(tabControl)            # Create a second tab
```

```
# tabControl.add(tab2, text='Graph')      # Add to window

# tab3 = ttk.Frame(tabControl)
# tabControl.add(tab3, text='Tab 3')

tabControl.pack(expand=1, fill="both")  # Pack to make visible
# Tab control End
    ------------------------------------------------------------


# Container for tabs
    ------------------------------------------------------------

t1c = ttk.LabelFrame(tab1, text='Search')
t1c.grid(column=0, row=0, padx=8, pady=4)

# t2c = ttk.LabelFrame(tab2, text='Extraction')
# t2c.grid(column=0, row=0, padx=8, pady=4)

# t3c = ttk.LabelFrame(tab3, text='Visualisation')
# t3c.grid(column=0, row=0, padx=8, pady=4)
# Containers End
    ------------------------------------------------------------------

# Tab 1 Content
    ------------------------------------------------------------------

inputLabel = ttk.Label(t1c, text="Web Search: ")
inputLabel.grid(column=0, row=0)

usersearch = tk.StringVar()
inputText = ttk.Entry(t1c, width=60, textvariable=usersearch)
inputText.grid(column=1, row=0)


# drop down lists
location = df['Place']
location = location.dropna()

LocationLabel = ttk.Label(t1c, text="Location: ")
LocationLabel.grid(column=0, row=1)

dropdownvarlocation = tk.StringVar(t1c)
dropdownlocation = ttk.OptionMenu(
```

```
    t1c, dropdownvarlocation, location[0], *location)
dropdownlocation.grid(column=1, row=1)
dropdownvarlocation.set('England')

amountLabel = ttk.Label(t1c, text='Search limit: ')
amountLabel.grid(column=2,row=1)

amount = [5, 100, 250, 500, 1000, 5000, 10000]
dropdownVarAmount = tk.StringVar(t1c)
dropdownAmount = ttk.OptionMenu(t1c, dropdownVarAmount,
    amount[0], *amount)
dropdownAmount.grid(column=3, row=1)
dropdownVarAmount.set(250)

TimeLabel = ttk.Label(t1c, text="Days: ")
TimeLabel.grid(column=4, row=1)

days = [7, 14, 21, 28]
dropdownvardays = tk.StringVar(t1c)
dropdowndays = ttk.OptionMenu(t1c, dropdownvardays, days[0],
     *days)
dropdowndays.grid(column=5, row=1)
dropdownvardays.set(14)

# Search google
my_api_key = "*"
my_cse_id = "*"
# This is the build phase
searchterms = df['Search']
searchterms = searchterms.dropna()


@threadpool
def twitter_search():
    location = dropdownvarlocation.get()
    number = dropdownvardays.get()
    today = (datetime.date.today())
    limit = (datetime.date.today() - datetime.timedelta(days
    =int(number)))
    searchamount = dropdownVarAmount.get()
    outputFileName = '{0}-results.tsv'.format(today)
    outputFile = codecs.open(outputFileName, "w+", "utf-8")
    outputFile.write(
        'username\tdate\tretweets\tfavorites\ttext\tgeo\
    tmentions\thashtags\tid\tpermalink\tClassification')
    for text in searchterms:
```

```
139         tweetCriteria = got.manager.TweetCriteria().
    setQuerySearch(
140             str(text)).setSince(str(limit)).setUntil(str(
    today)).setMaxTweets(int(searchamount)).setNear(str(
    location))
141         tweets = got.manager.TweetManager.getTweets(
    tweetCriteria)
142         print('Searching...\n')
143         print(str(text))
144         scr.insert(tk.END, '{0}\n'.format(text))
145 
146         for t in tweets:
147             decision = Classification.classify(t.text)
148             scr.insert(tk.END, 'Username : {0} Retweets :
    {1} \nTweet : {2} \nMentions : {3} # : {4}\n Decision:
    {5}'.format(
149                 t.username, t.retweets, t.text, t.mentions,
    t.hashtags, decision))
150             outputFile.write(('\n%s\t%s\t%d\t%d\t"%s"\t%s\t%
    s\t%s\t"%s"\t%s\t%s' % (t.username, t.date.strftime(
151                 "%Y-%m-%d %H:%M"), t.retweets, t.favorites,
    t.text, t.geo, t.mentions, t.hashtags, t.id, t.permalink
    , decision)))
152         outputFile.flush()
153         scr.insert(tk.END, '%d saved to file...\n' % len(
    tweets))
154 
155     print('Completed')
156 
157 
158 def google_search(search_term, api_key, cse_id, **kwargs):
159     service = build("customsearch", "v1", developerKey=
    api_key)
160     res = service.cse().list(q=search_term, cx=cse_id, **
    kwargs).execute()
161     return res['items']
162 
163 # Serch terms are entered here use num to edit returned
    value
164 
165 
166 @threadpool
167 def google_input(search_input):
168     results = google_search(
169         inputText.get(), my_api_key, my_cse_id, num=10)  #
    Change num for results returned.
```

```
    for result in results:
        r = pprint.pformat(result)
        # scr.insert(tk.END, r)
        # Could replace this print to a file as it outputs a JSON
    with open('search_result.json', 'w') as outfile:
        json.dump(results, outfile)
    google_json_parse()


extractedwords = []


def google_json_parse():
    with open('search_result.json', 'r') as f:
        readJson = pd.read_json(f)  # convert JSON to pandas dataFrame
        # scr.insert(tk.END, readJson) #inserts pulled webaddress to scrolling textbox
        urllist = readJson['link'].tolist()  # save Urls as a list
        # print(urllist)

        # below takes each webpage creates a new thread for each fetch and processing

        @threadpool
        def extraction(urllistinput):
            tcount = len(urllistinput)
            ccount = 1

            for item in urllistinput:
                scr.insert(tk.END, 'Now parsing {0}'.format(item))
                extractedwords.extend(
                    extract_nouns_websites.extract_nouns_from_webpages(item))
                # scr.insert(tk.END, extractedwords) # inserts extraction data to scroll textbox.
                # scr.delete(1.0, tk.END)
                scr.insert(
                    tk.END, '\n \n Completed {0} of {1}\n \n'.format(ccount, tcount))
                ccount += 1

        extraction(urllist)
```

```
207
208
209 # Graphing functions
210 def output_dictionary_with_names_and_their_counts(query):
211     return dict(Counter(query))
212
213
214 def flattened_list(input_list):
215     query = []
216     for item in input_list:
217         for sub_item in item:
218             query.append(sub_item)
219     return query
220
221
222 def values_dict(input_dict):
223     # Define a minimum number of occurrences for names to be
      considered
224
225     min_number_occurrence = 2
226     i = 0
227     list_keys = []
228     list_var = []
229     list_index = []
230     for key, var in input_dict.items():
231         if var >= min_number_occurrence:
232             list_keys.append(key)
233             list_index.append(int(i))
234             list_var.append(var)
235             i = i+1
236     return list_keys, list_index, list_var
237
238
239 def plot_bar(output_from_values_dict, query):
240
241     TOOLS = "crosshair,pan,wheel_zoom,box_zoom,reset,hover,
    save"
242     lists_dataframe = pd.DataFrame(
243         {'keywords': output_from_values_dict[0],
244          'index':  output_from_values_dict[1],
245          'occurrences': output_from_values_dict[2]
246          })
247     source = ColumnDataSource(lists_dataframe)
248     scr.delete(1.0, tk.END)
249     displaydf = lists_dataframe.drop(['index'], axis=1)
250     scr.insert(tk.END, displaydf)
```

```
251     #print(displaydf)
252     keyword_for_graph = lists_dataframe['keywords'].tolist()
253     occurrences_for_graph = lists_dataframe['occurrences'].
    tolist()
254 
255     p = figure(y_range=keyword_for_graph, plot_width=1000,
    plot_height=800,
256                tools=TOOLS, title="Occurrence of Keywords
    from the query input "+str(query))
257     p.xaxis.axis_label = 'Occurrence of Keywords'
258     p.yaxis.axis_label = 'Keywords'
259     p.hbar(y=keyword_for_graph, right=occurrences_for_graph,
260            left=0, height=0.5, fill_alpha=1)
261 
262     output_file(
263         "query.html", title="Occurrence of Keywords from the
     query input " + str(query))
264     show(p)
265 
266 
267 # Buttons 1 and 2 - onClickSearch for pre built search --
    onClickUser for custom search
268 
269 def _onClickSearch():
270     scr.delete(1.0, tk.END)
271     # scr.insert(tk.END, main_extraction_file.
    get_nouns_from_website('https://epsrc.ukri.org/funding/
    applicationprocess/routes/newac/nia/'))
272     google_input(inputText.get())
273     graphBtn.config(state='active')
274 
275 
276 def _onClickSearchtwitter():
277     scr.delete(1.0, tk.END)
278     twitter_search()
279 
280 
281 def _onClickGraph():
282     input_dict = flattened_list(extractedwords)
283     # print(output_dictionary_with_names_and_their_counts(
    input_dict))
284     plot_bar(values_dict(
285         output_dictionary_with_names_and_their_counts(
    input_dict)), input_dict)
286 
287 
```

```
submitBtn = ttk.Button(t1c, text='Web Search', command=
    _onClickSearch)
submitBtn.grid(column=6, row=0)

submitBtntwitter = ttk.Button(
    t1c, text='Twitter Search', command=
    _onClickSearchtwitter)
submitBtntwitter.grid(column=6, row=1)

graphBtn = ttk.Button(
    t1c, text='Graph', command=_onClickGraph, state='
    disabled')
graphBtn.grid(column=5, row=3)

# Scrolling text box
    ------------------------------------------------------------

scrolW = 120
scrolH = 35
scr = scrolledtext.ScrolledText(t1c, width=scrolW, height=
    scrolH, wrap=tk.WORD)
scr.grid(column=0, row=2, sticky='WE', columnspan=7)
# Scrolling text box END
    ----------------------------------------------------

# Lower buttons
    ------------------------------------------------------------------------



def _onClickClear():
    scr.delete(1.0, tk.END)


clrBtn = ttk.Button(t1c, text='Clear', command=_onClickClear
    )
clrBtn.grid(column=6, row=3, sticky='E')

# Drop down
    ------------------------------------------------------------

def trainModel():
    Classification.build()

def _quit():
    url = inputText.get()
```

```
322     # regex to grab domain name for file save
323     domain = url.split('//')[-1].split('/')[0]
324     if os.path.isfile('query.html'):
325         if not os.path.exists('./archive'):
326             os.makedirs('./archive')
327         # adds the domain name to the file save
328         shutil.copy2('query.html', './archive/query-{}.html'
    .format(domain))
329         os.remove('query.html')
330     else:
331         print('No File to remove')
332     win.quit()
333     win.destroy()
334     exit()
335 
336 
337 # Creating a Menu Bar
338 menuBar = Menu(tab1)
339 win.config(menu=menuBar)
340 
341 # Add menu items
342 fileMenu = Menu(menuBar, tearoff=0)
343 #fileMenu.add_command(label="New")
344 fileMenu.add_command(label="Train model", command=trainModel
    )
345 fileMenu.add_separator()
346 fileMenu.add_command(label="Exit", command=_quit)
347 menuBar.add_cascade(label="File", menu=fileMenu)
348 
349 
350 def _msgBoxAbout():
351     mBox.showinfo('About', 'V:0.0.3')
352 
353 
354 # Add another Menu to the Menu Bar and an item
355 helpMenu = Menu(menuBar, tearoff=0)
356 helpMenu.add_command(label="About", command=_msgBoxAbout)
357 menuBar.add_cascade(label="Help", menu=helpMenu)
358 # Menu Bar END
359 
360 # Main loop
361 
362 win.mainloop()
```