Aston University

Some pages of this thesis may have been removed for copyright restrictions.

If you have discovered material in Aston Research Explorer which is unlawful e.g. breaches copyright, (either yours or that of a third party) or any other law, including but not limited to those relating to patent, trademark, confidentiality, data protection, obscenity, defamation, libel, then please read our Takedown policy and contact the service immediately (openaccess@aston.ac.uk)

Enhancing the Interactivity of a Clinical Decision Support System by using Knowledge Engineering and Natural Language Processing

Mohammed Ashrafull Islam

Doctor of Philosophy

Aston University

June 2018

©Mohammed Ashrafull Islam, 2018.

Mohammed Ashrafull Islam asserts his moral right to be identified as the author of this thesis.

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without appropriate permission or acknowledgement.

Acknowledgements

I would like to express my sincere gratitude to my advisor Dr. Christopher Buckingham for the effort he put into guiding me throughout this work, for his patience, motivation, and immense knowledge.

Many thanks to the GRiST team members for delivering such a great project. I would also like to thank my work colleagues for their kind support.

I would like to dedicate this work to my late father and to my mother for their unconditional love and inspiration. I would not have been able to endure these years without the love and support of my family, siblings and in-laws. A great big thank you to my three little boys Zakariya, Yusuf and Saleh for postponing so many fun things until I finished my dissertation. Above all, I lovingly thank my wife for all her support, understanding, and continued help over the years.

Abstract

Mental illness is a serious health problem and it affects many people. Increasingly, Clinical Decision Support Systems (CDSS) are being used for diagnosis and it is important to improve the reliability and performance of these systems. Missing a potential clue or a wrong diagnosis can have a detrimental effect on the patient's quality of life and could lead to a fatal outcome. The context of this research is the Galatean Risk and Safety Tool (GRiST), a mental-health-risk assessment system. Previous research has shown that success of a CDSS depends on its ease of use, reliability and interactivity. This research addresses these concerns for the GRiST by deploying data mining techniques. Clinical narratives and numerical data have both been analysed for this purpose.

Clinical narratives have been processed by natural language processing (NLP) technology to extract knowledge from them. SNOMED-CT was used as a reference ontology and the performance of the different extraction algorithms have been compared. A new Ensemble Concept Mining (ECM) method has been proposed, which may eliminate the need for domain specific phrase annotation requirements. Word embedding has been used to filter phrases semantically and to build a semantic representation of each of the GRiST ontology nodes.

The Chi-square and FP-growth methods have been used to find relationships between GRiST ontology nodes. Interesting patterns have been found that could be used to provide real-time feedback to clinicians. Information gain has been used efficaciously to explain the differences between the clinicians and the consensus risk. A new risk management strategy has been explored by analysing repeat assessments. A few novel methods have been proposed to perform automatic background analysis of the patient data and improve the interactivity and reliability of GRiST and similar systems.

Keywords: Clinical Decision Support Systems, Concept Extraction, Risk Classification, Information gain, Word embedding, Galatean Model, Decision Tree, SNOMED-CT

Table of Contents

List of Tables

List of Figures

List of Abbreviation

A

C

D

F

G

H

I

L

M

N

Q

S

U

V

W

X

1 Introduction

1.1 Introduction

Clinical Decision Support Systems (CDSSs) can improve both patient care outcomes and reduce the cost of care (Berner & La Lande, 2007). CDSSs are interactive expert systems, which use embedded clinical knowledge to help health professionals analyse patient data and make decisions regarding diagnosis, prevention, and the treatment of health problems (Wu, Lu and Duan, 2008). A systematic review of CDSSs found that over 90% of the systems significantly improved clinical care (Kawamoto, Del Fiol, Lobach and Jenders, 2010). They have shown great promise and contributed towards reducing medical errors and improving patient care (Kawamoto, Houlihan, Balas and Lobach, 2005).

Despite increasing emphasis on CDSS in improving care and reducing costs, evidence supporting its widespread use is limited (Bright et al., 2012). The acceptance of the widespread applications of CDSS is hampered by factors such as complexity of the system, time consuming for the doctors and lack of decision accuracy (Al-gamdi, 2014). The GRiST is a CDSS for mental health risk assessment. This research uses GRiST as a test case and proposes new methods to improve the interactivity and accuracy of the GRiST or similar expert systems.

Firstly, we looked at how we can use textual data to identify risk and alert the clinician. Secondly, how we could use potential relationship among attributes (GRiST ontology nodes) and identify risk at an early stage of the assessment. Thirdly, we looked at how we can validate the clinical judgement and provide feedback. Application of these techniques may improve the overall performance of the GRiST by making it more interactive and increase its acceptance among clinicians. This in turn can enhance patient safety.

Key phrase extraction is an important first step for Natural Language Processing (NLP) tasks. We have reviewed many well-known phrase extraction methods. Many of these phrase extraction systems need training with domain specific data and some of them

can extract key phrases; however, those phrases may not be domain relevant. To overcome this problem, we propose a two-stage generic method. At the first stage, it extracts key phrases by using linguistic patterns and then at the second stage those extracted phrases are filtered for semantic similarity with the domain by using word embedding. We have shown that our method can perform better than existing well-known key-phrase extraction systems. Many other exploratory semantic analysis tasks have been carried out which may help with future NLP research.

In the existing literature, there are encouraging mentions of predicting diseases from clinical notes. We have reviewed and applied many of the existing text classification methods with our dataset. A comprehensive number of experiments have been carried out from various perspective to predict suicide risk from the clinical narratives. We have experienced many similar difficulties as previous researchers. Predicting different levels of risk was challenging. Using full-text data, extracted phrases, document embedding, etc. all methods have been explored, and their performances were critically reviewed.

Electronic health records provide better value for clinicians by allowing clinicians to reliably identify adverse events (Jha, 2011). High suicide risk incidents are rare in the GRiST dataset and identifying them by using classification methods is challenging due to the class imbalance problem. We assume that there exist some non-linear relationships among GRiST ontology nodes, which may affect the risk judgement. It would be particularly important to help identify which of these potential relationships causes the risk to be higher. Empirical evidence suggests that statistically related nodes appear more in high-risk category patients.

Frequent itemset mining as introduced by Agrawal, Imielinski, & Swami (1993) is generally used for market basket analysis to predict users shopping habits. The technique is also applied in finding disease symptom relationships in literature. We have applied frequent itemset mining for detecting high-risk patients within GRiST dataset. This has the benefit of not needing to gather all the data prior to a calculated prediction being made. The itemset (symptoms) related to higher risks of suicide are found to be rare. We have proposed a new approach to overcome this problem and achieved better accuracy than simple frequent itemset mining.

The third approach to improve the GRiST system was to assess the reliability of clinical judgements and provide interactive feedback to improve accuracy. For this, we have used total information gain or relative weights of an assessment. This is a novel method to identify the difference between a clinician given and calculated risk, explain the probable reason and guide clinicians to make a better judgement. We have found that the accuracy of the clinical judgements depends on the total information gathered by the clinicians. The proposed method could notify clinicians about the accuracy of their risk judgement and provide feedback to improve it.

The focus of this research was to identify patterns in the GRiST data and knowledge structure to make the GRiST system more interactive and user-friendly. Finding clues in the comments, detecting node relationships and explaining the risk differences is a significant contribution to the GRiST project. The methods described in this research can easily be applied to other CDSS like GRiST.

1.2 **Ethical Approval**

Ethics of the research is covered by the GRiST project. The patient data has been kept confidential and secure at all the times. No data has been shared with any other third party. The following is the ethics clearance for analysing the GRiST database.

Title of the Database: Analysing the Galatean Risk and Safety Tool (GRiST) Database
REC reference: 13/EM/0007
IRAS project ID: 119801

REC is the Research Ethics Committee and IRAS is the submission form for the ethics application.

1.3 Research Aim

The overall aim of this research was to analyse the GRiST data and its ontological structure to find patterns within them and enhance the system with dynamic background analysis, improve user interactivity and validate the risk judgement. The findings could also be applied to other similar systems in the future.

1.4 Research Questions

The context of this research is the GRiST system in which the data is inputted as both free text comments and numerical values. The clinical narratives can hold some clues that numerical data may be missing. Ontology node interactions may also provide useful information. It is a challenge to combine all these clues and build a comprehensive understanding of a patient's mental health.

This research has attempted to answer the following questions:

1. *How can NLP technology be used to extract concepts from clinical comments to represent a GRiST node or a patient?* We were particularly interested in comparing and extending existing unsupervised methods.

2. *How can phrases be stemmed by semantic similarity and how does it compare with string-based similarity?* Generally, we can find a base form of a word by stemming. Stemming algorithms mostly work on the prefix or suffix variations of a word. Two words may appear completely different by character matching but semantically they might be very close. We may use semantic vectors to do semantic stemming and use these stemmed phrases for risk prediction.

3. *Can semantic vector representation of GRiST nodes help us to identify any patterns that may assist us in improving the overall GRiST system?* We can build

the semantic vector of a node or an assessment by using its constituent words vector. This may assist in risk analysis.

4. *How does the data in the GRiST and its ontological structure relate to other ontology like SNOMED-CT and the implication of these relations on suicide risk?*

5. *How do risk predictions produced by using raw text, extracted phrases, word vectors and numerical data compare with each other?* Risk calculated by alternative methods may assist in validating the clinicians given risk.

6. *How can statistical measures such as chi-square, or itemset mining such as fp-growth, be used to find relationships between the GRiST nodes? How does the presence of these relationships might affect risk judgement?*

7. *Could the difference between the clinician given and calculated risk be explained by identifying patterns in the raw data, particularly by using information theory?* If the clinician given risk is higher, or it differs significantly from the calculated risk, then it would be extremely useful to know the probable reasons for this. Knowing the answer to this question may also help us to take the necessary measures to mitigate this exceptional circumstance.

1.5 Contributions to Knowledge

The context of this research was the GRiST clinical decision support system but many of the findings and techniques can be applied to the field of natural language processing and expert system design. The following points are a summary of the contributions made by this research:

1. Previous researchers have described various methods to filter phrases such as by frequency (Pudota, et al. 2010), latent semantic analysis (J. Chen et al.,2006) or concept graph (Bleik, Xiong, Wang, & Song, 2010). We have developed an ensemble concept mining (ECM) method, which can automatically extract

domain specific key phrases from the text. Our method extends the previous methods by including word vector based semantic filtering. Empirical findings show that the utility of a complex method is not significantly higher than the simple n-gram phrase extraction method followed by a semantic filtering. The greatest benefit of our method is that it does not require human annotation. For the i2b2 dataset, our approach gives better results than the Rake or OpenNLP approaches.

2. The distributional hypothesis in linguistics describe that the meaning of a word can be determined by the company it keeps (Firth, 1957). Vector space model (VSM) uses word co-occurrence counts from large corpora to represent lexical meaning (Padó & Lapata, 2007). In VSM, words are represented by vectors. Two words would be closer if their meaning is similar. We have extended this idea and proposed that within a document a concept word or phrase will have lots of other words or phrases that are semantically related. We proposed a method to extract domain relevant key phrases by scoring its semantic relatedness to other words in the same document or corpus. This is a simple but fully automatic method that can save time and cost in finding relevant key words.

3. Methods for detection of suicide and other risks from clinical notes have been described by researchers for example (Thompson, Bryan, & Poulin, 2014), (O'Dea et al., 2015), (Yang, Spasic, Keane, & Nenadic, 2012), (Pestian et al., 2008) and many others. These methods include machine learning, rule-based, statistical text mining (STM), and text weighting approaches (McCart et al., 2012). We have reviewed different approaches with our dataset and additionally used document embedding and GRiST ontological structures. Use of a bigger dataset and the comparative study provides valuable information for future research.

4. Associated rule mining is being used to determine the relationships among symptoms and to predict disease. Previously researchers such as (Lacković, de Carvalho, Zhang, & Magjarević, 2014) and (Huang, Huang, Chen, Liu, & Huang (2012) described the application of frequent itemset mining to discover relationships among symptoms and mental health illness. We have used the FP-growth algorithm to predict high-risk category patients. Like others, we have

found that high-risk categories are rare events and normal support and confidence measures do not provide better results. We have proposed a new method by which we can apply multiple rules and achieve better accuracy in predicting high-risk patients.

5. Entropy represents the randomness of a system and information gain identifies the predictive power of an attribute. We have used the notion of information to quantify the reliability of a clinical assessment. It was assumed and subsequently observed in the GRiST data that a clinical judgement is closer to the calculated risk if it has collected more information. We have proposed a novel method to determine the probable reliability of an assessment from the information it has collected. Our method can also provide feedback to improve assessment accuracy. The technique can make a system like GRiST more dynamic and truly interactive.

The following are some minor but useful additional contributions of this research:

6. This research describes a simple technique for using dependency relationships between words to find similar words. It may be used instead of using the spanning window based approach that is generally used in distributional word embedding.

7. We have developed an algorithm to dynamically select attributes and run predictive calculations when some data may be missing. In the context of the GRiST data, the predictive performance of this technique was found to be better than before.

8. We proposed a semantic stemming technique based on the word2vector to reduce the number of extracted phrases. This can be used in conjunction with other methods such as the Levenshtein distance to reduce the list of key phrases. A method has been described to create a list of semantically relevant phrases for a GRIST node automatically.

9. A new method has been proposed to assist clinicians to adopt a better risk management strategy. This method uses previously completed repeat assessment data available in GRiST; it measures the increase or decrease of a symptom (attribute) over time and the implication of the symptom on suicide risk. Based on these two pieces of information it can provide feedback to the clinician to adopt a better risk management strategy.

1.6 Contributions to Practitioners

The outcome of this research could help clinical practitioners to make a better risk assessment. The following are some of the key benefits for the clinical practitioners.

1. The clinician could get early feedback / be alerted to the potential suicide risk related symptoms before the assessment is completed. The proposed automatic concept extraction method and SNOMED-CT mapping can help in this regard.

2. When a patient carried out a self-assessment using GRiST, clinicians might get feedback including the potential suicide risk automatically calculated by the proposed pattern mining method.

3. GRiST node association rules (relationships) can be used to alert clinicians in real-time about the potential suicide risk.

4. The system could provide feedback on the final risk judgement and provide specific guidance to make any possible amendments based on the proposed information gain method.

5. The proposed hard and soft node analysis method can provide feedback to choose the best options for clinical intervention.

6. Senior management can validate the risk judgement based on the proposed information gain method and may review a selective set of assessments.

1.7 Software Tools Developed

In the course of our research, many software tools have been developed. Some of these tools are built with other relevant open source APIs. These tools will be released as open source software in the foreseeable future.

7. A simple Java based tool has been developed, which can return a SNOMED-CT coded XML output for an inputted sentence. This allows other scripting languages such as PHP to use it as a web service.

8. A Java based web service for the Stanford parser was developed. Given a single sentence, it can output a Stanford parse tree. A similar service is available from Stanford online, but if data is confidential then our tool may be more useful as it can be run locally.

9. An ensemble phrase filtering algorithm has been developed and implemented in both PHP and Java.

10. A very fast C language based REST service has been developed that can output a word vector for a given word.

11. Many other utility tools such as SNOMED-CT and WordNET browsers have been created, which may assist other researchers. Unlike others, these tools use a database as a storage system.

12. A script to find similar words based on dependency relationships has been developed.

13. Scripts for calculating chi-square, PMI and Information gain have been implemented for the GRiST data.

14. A script for filtering the FP-growth extracted frequent patterns and using them for risk prediction has been implemented.

1.8 Report Organisation

The remainder of this report is organised as follows:

Chapter 2: Background and Literature Review
This chapter reviews existing literature and research methods. It describes some of the existing clinical CDSS systems and their working principles. Previous works that are relevant to our research activities are described in detail. How our research extends the previous research works is described in the respective subsections.

Chapter 3: The GRiST CDSS
The data for this research came from the GRiST system and it is the primary focus of this research. A detailed description of GRiST and its working principles are given in this chapter. The Galatean model is also described in detail with examples, as this is the underlying model of the GRiST system.

Chapter 4: Concept Extraction
This chapter describes recent research in phrase extraction and concept extraction. Different phrase extraction methods were applied to the GRiST data and the results were critically reviewed. This chapter also describes our proposed ensemble concept mining (ECM) method.

Chapter 5: Semantic Processing (Exploratory)
This chapter explores different ways to represent GRiST nodes and patients semantically. We have described the method to semantically connect GRiST nodes based on the text they share. It also mapped each GRiST node to the SNOMED-CT

concept. This is an exploratory work to find possible patterns and improve our understanding of the GRiST dataset.

Chapter 6: Risk Prediction and Classification

In this chapter, different text classification algorithms are used to predict suicide risk. Raw texts, extracted phrases and semantic vectors etc. have all been used to predict risk. The performance of these different methods was critically analysed. A method to improve regression analysis is also described. How the GRiST nodes value changes in repeat assessments is discussed, this could be particularly useful for risk management purposes.

Chapter 7: Association Rule Mining

Literature review shows that the use of frequent itemset mining to identify disease symptoms relationships is increasing. We aim to identify high-risk patients based on the frequently occurring relationships among GRiST nodes and suicide risk. The interaction between GRiST nodes is described using the chi-square method. The FP-growth method was used for mining frequent itemset. A new method is proposed to identify the high-risk category patients effectively from the GRiST data.

Chapter 8: Reliability of Risk Judgement

The difference between a clinician given risk and a calculated risk needs to be as small as possible. In risk calculation, the relative contribution of different GRiST nodes can vary. The reliability of a risk assessment may depend on how much information is collected by the clinician. This chapter explains various methods to determine the amount of total information collected by the clinician and uses them to explain the reliability of an assessment.

Chapter 9: Conclusion

This chapter gives a concluding critical analysis of each of the activities and findings followed by suggestions for future improvements. The practical implications of this research on the GRiST system as well as on any other similar CDSS system are discussed.

2 Background and Literature Review

2.1 Introduction

The domain of this research is mental health risk assessment using the clinical decision support system (CDSS). This chapter briefly describes the mental health problems and importance of its assessment. We also discuss the different methods of mental health risk assessments currently in use. Popular CDSS systems have been reviewed and their underlying methods have been compared. Finally, literature related to our research, limitations in the existing methods and the formulation of our own research activities are discussed in their respective sections.

2.2 Mental Health Problems

2.2.1 The Impact of Mental Illness

According to the Mental Health Foundation (2015), one in four people in the UK will experience a mental health problem in any given year. It states that mental health problems are one of the main causes of the burden of disease worldwide. In the UK, they are responsible for 28% of the total burden of disease compared to 16% each for cancer and heart disease. Since 2009 the number of working days lost to stress, anxiety and depression has increased by 24% and those lost to serious mental illness has doubled (Bridges, 2014).

“Psychosis is characterised by hallucinations, delusions and a disturbed relationship with reality, and can cause considerable distress and disability for the person and their family or carers” (NHS England, 2016, p.6). In 2013 there were 6,233 suicides recorded in the UK for people aged 15 and over. Of these, 78% were male and 22% were female

(Mental Health Foundation, 2015). Mental health problems can have a serious impact on an individual's quality of life. This can also impact the people around them.

The Mental Health Foundation (2015) claim that mental health services in the UK are overstretched, they have long waiting times and, in some regions, lack specialist services. There is a huge treatment gap in mental health care in England, with about 75% of people with mental illness receiving no treatment at all (Davies, 2013). In the UK, the estimated cost of mental health problems is roughly £70-100 billion each year and accounts for 4.5% of GDP (Mental Health Foundation, 2015).

There are strong links between physical and mental health problems. Research has found that 30% of people with a long-term physical health problem also have a mental health problem and 46% of people with a mental health problem also had a long-term physical health problem (Mental Health Foundation, 2015). A number of reviews and studies have found that the lifespan of people with severe mental illness (SMI) is shorter compared to the general population and this is attributed to the likelihood of them having a physical illness (DE Hert et al., 2011). Self-neglect or self-harm could also lead to the poor physical health conditions.

2.2.2 The Importance of Risk Assessment

Mental illness can affect anybody at any age and it can have a significant impact on an individuals' quality of life, their family and community (Davies, 2013). Not only the human costs associated with mental illness but the economic burden it also imposes is significant (Bridges, 2014). Research shows that early intervention in psychosis produces better clinical outcomes and is also more cost-effective (Singh, 2010).

A report by NHS England (2016) mentions that people who experience psychosis can and do recover. The time from the onset of psychosis to the start of evidence based treatment could significantly influence the ultimate long-term outcomes. The sooner treatment is started the better the outcome and the lower the overall cost of care (NHS England, 2016). Early intervention can help to lessen the impact of the condition, reduce the risk of further (and often more debilitating) episodes and increase the possibility of

better social and functional outcomes, such as completing education and staying in employment (McDaid et al., 2016).

According to McDaid et al. (2016) better outcomes are driven by three key components: (1) well-engaged health and other sector professionals, working collaboratively to achieve long-term goals; (2) infrastructure supportive of early intervention services; and (3) the development of early intervention care pathways for people with psychosis and their families.

In this context, having a tool like GRiST available to the clinicians and service users may provide an opportunity for early intervention, which is more cost-effective. A risk assessment tool should be based on a systematic approach that is proven by research. The next section describes some of the risk assessment approaches in detail.

2.3 Approaches to Risk Assessment

Previously, risk assessment was more focused on prediction but currently risk assessment systems have attempted to unite research evidence with clinical practice and have begun to incorporate aspects of risk management (Bouch, 2005). In other words, the focus has now shifted from prediction to prevention. There are three broad approaches to risk assessment (Bouch, 2005): clinical, actuarial and structured professional judgement. The following is a short description of these three approaches adopted from (Bouch, 2005).

2.3.1 The Clinical Approach

In the clinical approach, decisions are made on the basis of a clinicians' judgement. This judgement is based on evidence, but it is also subjective, intuitive and informed by the experience of the clinician. In suicide risk assessment, decisions are made about supervision, treatment and hospitalisation on the basis of professional opinion. Such

decisions may be criticised as being based on the feelings of the clinician rather than the evidence (Bouch, 2005).

2.3.2 The Actuarial Approach

The actuarial approach to risk assessment has been developed to meet the concerns of the clinical judgement. This approach uses formal, algorithmic methods and follows objective procedures for classifying risk. The ultimate objective is to calculate a risk probability of a future outcome. For example, patient A has a 50% chance of committing a violent act in the next 2 years. But this does not inform clinicians about the circumstances, severity or imminence of the act in question. The risk statement about patient A may be mathematically correct but has limited usefulness in informing management, especially in the short term (Bouch, 2005).

2.3.3 Structured Professional Judgement

Structured professional judgement (SPJ) is an approach to risk assessment where evidence base for risk factors are combined with the individual patient assessment. It assists but does not replace psychiatric opinion. Clinicians make a structured assessment for the formation of a risk management plan. This facilitates teamwork among multidisciplinary teams. Following a structure supports evidence-based practice, and also increases the transparency of decision making for the purpose of clinical governance (Bouch, 2005).

According to Petrik, Gutierrez, Berlin, & Saunders (2015) qualitative analysis produced six themes that impact suicide risk assessment. They are time, privacy, collaboration, consultation with other professionals and integration of a standard screening protocol in routine care and systemic themes. Patient engagement in the assessment process and the providers' communication approach with the patients and other providers can affect the effectiveness of suicide risk assessment (Petrik et al., 2015).

A comparison between actuarial and SPJ in the case of 177 adjudicated juvenile offenders provides moderate support for the continued use of the SPJ framework (Childs, Frick, Ryals, Lingonblad, & Villio, 2014). Most of the differences in these two methods have been found to happen in higher risk categories. SPJ methods include a mix of static and modifiable factors, and rather than considering them as classification procedures but as prognostic models are more appropriate (Falzer, 2013).

Having a structured system like GRiST can help to follow a standard procedure for all the patients in an efficient manner. GRiST collects information in a structured manner and also records clinician judgements. This research has attempted to explain the risk differences between clinicians and calculated risk by using information gain measures.

According to Shortliffe & Cimino (2014), clinical decision support systems can be clustered into three different types:

1. Information Management Systems, for storing and retrieving clinical knowledge. The interpretation of the stored knowledge is left to the clinician.

2. Focusing Attention Systems, which alert the user to possible conflicts or problems that might have been missed.

3. Patient Specific Recommendation Systems, which provide a personal assessment of a patient, usually following simple logical rules.

2.4 Clinical Decision Support Systems

CDSS provides clinicians, staff, patients, and other stakeholders with domain knowledge and intelligently filtered patient specific information presented at appropriate times, to enhance healthcare (Osheroff et al., 2007). CDSS may provide a standardised facility for risk assessment and management.

Many clinical decision support systems are in use in the biomedical sector and their use is expected to rise in the United States due to the Health Information Technology for Economic and Clinical Health (HITECH) Act, which stipulates that health care providers must demonstrate the meaningful use of health IT by 2015 (Rouse, 2010). In this chapter, we have reviewed some of the prominent CDSSs along with their technical details, which are relevant to our research.

2.4.1 Example of CDSS

Before developing a methodology for GRiST improvement, it is important to know more about other CDSSs currently in use. The following table shows some of the prominent CDSSs along with a brief technical detail of those that are relevant to our research.

Table 1 Clinical Decision Support Systems and their methods

System Name	Description and Methods
IMASC	Intelligent MultiAgent System for Clinical Decision Support (IMASC) developed by Czibula, Czibula, Cojocar, & Guran (2008) uses a central database to store symptoms and makes predictions based on them. The system learns of new symptoms from the clinician's feedback.
ZynxEvidence	ZynxEvidence is an online resource that provides best practice guidance for physicians, nurses, and health professionals and also displays evidence-based clinical content in a hospital setting. The content is divided into more than 145 modules, which addresses clinical conditions, procedures, and patient problems (ZynxHealth, 2012).
CADUCEUS	CADUCEUS (also known as Internist, QMR), was designed at the University of Pittsburgh and was considered to have an extensive knowledge base, of more than 750 disorders and almost 4,500

	interrelated findings or disease manifestations. But to return a diagnosis a disease needed to be in the system. Updating the knowledge base became a huge task and it was last updated in 2001 (Moore and Loper, 2011).
MYCIN	MYCIN developed by Shortliffe (1977) was a very early (1970) expert system, which used goal focussed reasoning in an IF – THEN method to search its knowledge base. MYCIN could ask additional questions about the patient, suggest appropriate testing, offer possible diagnosis and recommend a course of treatment (Moore and Loper, 2011).
Iliad	Iliad was developed at the University of Utah. It uses the frame-based version of Bayesian reasoning to calculate the posterior probabilities of various diagnoses under consideration, given the findings present in a case. Iliad which was developed primarily for diagnosis in Internal Medicine, which now covers about 1500 diagnoses in this domain, based on several thousand findings (OpenClinical, 2001).
Isabel	Isabel is a web-based diagnosis checklist system. Physicians enter age, gender and clinical features, either by free text or taken directly from an electronic medical record and Isabel instantly returns a list of possible diagnoses. The Isabel engine is powered by statistical natural language processing technology and is updated continually from medical textbooks and journals with 3 separate and proprietary taxonomies (Isabel, 2012).
Watson	Recently the IBM Watson system became very well known. Watson combines natural language processing, dynamic learning, and hypothesis generation and evaluation to give direct, confidence-based responses (High, 2012).
DiagnosisPro	Diagnosis and differential diagnosis of more than 11 thousand

	diseases and 30 thousand medical conditions. (Shahsavarani, Abadi, Kalkhoran, Jafari, & Qaranli, 2015). Method: Knowledge-based.
Dxplain	Diagnosis and differential diagnosis of internal diseases, educational application. Method: Knowledge-based, pseudo-probabilistic algorithm, Bayesian logic, (Shahsavarani et al., 2015)
ESAGIL	Diagnosis of diseases according to signs and symptoms, blood and urine test. Method: Knowledge-based, dissociative reasoning (Shahsavarani et al., 2015)
Litmusdx	Diagnosis and differential diagnosis of 11 thousand diseases, presentation of 300 therapeutic protocols, presentation of 50 thousand medicines, 200 thousand medicine usage cautions, medical test interpretations, medical files. Method: Knowledge-based (Shahsavarani et al., 2015)
Clinical Rules	Medicine prescription, consumption monitoring. Knowledge-based, Knowledge management, Clinical Rules Engine, G standard MFB, Andere protocollen (Shahsavarani et al., 2015)
SimulConsult	Diagnosis of 5300 diseases especially genetic and neurological. Method: Knowledge-based, Bayesian inference engine, bioinformatics genome annotation, statistical pattern-matching approach (Shahsavarani et al., 2015).

There are many other CDSS systems in use, more information about them can be found in (Pawar and Patil, 2012), (Bright, Wong and Dhurjati, 2012) and (Curtain & Peterson, 2014). The underlying methodologies of different systems vary widely. In the next section, we have discussed methodologies that are used in mental health domain.

2.4.2 CDSS for Mental Health

The following are some of the well-known clinical decision support systems that are being used in mental health risk assessment.

Table 2 Example of CDSS in the mental health domain

System Name	Description and Methods
OQ Analyst	The OQ Analyst utilizes the Outcomes Questionnaires (OQ) in electronic form to track general distress among patients.
Carepaths	The Carepaths system uses the OQ Analyst functions along with other disease-specific scales and integrates them into an online mental health electronic medical record offering.
Q-logic	The Q-logic system uses the Brief Symptom Inventory (BSI-18) for its main outcome measure.
CRMT: Clinical Risk Management Tool/Working with Risk	The CRMT is a structured template checklist of relevant risk and contextual factors. The tool includes a structured assessment of suicide, neglect, violence and other risks
FACE: Functional Analysis of Care Environments	FACE is a portfolio of assessment tools designed for adult and older people's mental health settings. Five sets of risk indicators are coded as present or absent and then a judgement of risk status (0–4) is given.
RAMAS: Risk Assessment Management and Audit Systems	RAMAS consists of a framework and a set of structured professional judgement tools designed to improve quality and safety in mental health care.
START: Short-term Assessment of Risk and Treatability	START is a risk assessment and management decision support system developed in Canada. The service user's strengths and risks on each of 20 dynamic factors are assessed on a scale of 0–2.

The above information is summarised from (Brown, 2012) and (Department of Health, 2007). In comparison to others, GRiST uses its own Galatean risk assessment model, which is discussed in Chapter 3.

2.4.3 CDSS and NLP Technology

According to Friedman (2005) the design of NLP based clinical decision support systems faces many issues such as availability of clinical text, confidentiality, interoperability, the expressiveness of the natural language, and abbreviated medical text. Over the last two decades, there have been efforts to develop biomedical NLP systems for mining information from clinical narratives and mainly two methods, rule-based and machine learning based have been used (Liu, Weng and Yu, 2012). The two approaches are described in the following subsections.

2.4.3.1 Rule-Based Approach

The rule-based system uses rules to make deductions or choices. Comprehensive syntactic or semantic knowledge rules are usually applied to extract encoded information from clinical narratives (Liu, Weng and Yu, 2012). Some rule-based systems are shown below:

Table 3 Rule-based CDSS example

System Name	Technology	Description
MedEx	rule-based	Extracting medication and related fields from text.
MedLEE	rule-based	Process clinical information expressed in natural language.
MERKI	rule-based	Extract medication names and the corresponding attributes.

Table 4: Rule-Based Approach

2.4.3.2 Machine Learning-Based Approach

Machine learning based systems learn from their data source by using various statistical and artificial intelligence algorithms (Liu, Weng and Yu, 2012). Some machine learning-based systems are shown below:

Table 5 CDSS system and their technology

System Name	Technology	Description
AskHERMES	SVM (support vector machine)	Retrieves and mines large sets of literature documents and clinical notes pertaining to specific questions.
Lancet	CRF (conditional random field)	Automatically extracts medication events consisting of medication names and their prescribed use.
NegScope HedgeScope	CRF	Detect negation and hedge cues as well as their scopes in both the biomedical literature and clinical notes.
SymText	Bayesian network	Extract pneumonia-related findings from chest radio-graph reports.

Table 6: Machine Learning based approach

The above two tables are created by summarising the information provided in (Liu, Weng and Yu, 2012).

2.4.4 CDSS and Knowledge Representation

Constructing the Knowledge Base (KB) of a clinical decision support system is an important task that determines the success of the system (Stojkovska, Loskovska, & Member, 2010). Medical knowledge can be acquired and stored in many different ways for later use in a computerised system. A detailed review of the medical knowledge

acquisition and representation methods in CDSS has found the following common methods (Stojkovska et al., 2010):

Logical conditions: These are generally Boolean logic to check if a variable is within or outside of a bound. For example, “is the patient’s heart rate below 50 BPM?”. This is mainly used for alerts.

Rules: Contains IF-THEN rules. Reasoning process can chain together rules until a decision is reached. An example of CDSS using this method is MYCIN.

Graphs/Networks: Decision trees and artificial neural networks allow graphical representation of the medical knowledge. An example of CDSS using this method is DXplain.

Structural representations: Knowledge is stored in a well-organised structure such as ontology. An example of CDSS using this method is CENTAUR.

In GRiST, knowledge is represented using an ontology created by experts. The ontology creation process is discussed later in Chapter 3.

2.4.5 CDSS and Reasoning Methods

The reasoning is a fundamental task of the inference engine of a clinical decision support system, which is performed by combining medical knowledge with patient specific data and makes appropriate decisions (Aleksovska-Stojkovska & Loskovska, 2010). They have highlighted the following common methods of reasoning from the literature:

a) Rule-based reasoning is based on “if-then-else” rule statements.

b) A case-based reasoning searches for commonly occurring patterns that match the various stored cases.

c) Model-based reasoning provides a framework for diagnosing an artefact by comparing its behaviour with a model.

d) Bayesian reasoning is based on conditional probabilities. It predicts the probability of an event based on other events.

e) Heuristic reasoning methods exploit the information processing structure of the reasoning system and find reasonable answers.

f) A semantic network is a graphical representation of interconnected nodes that can be used to support automated systems for reasoning.

g) Neural networks are a black box modelling technique that model relationships by learning from historical data and pattern recognition.

h) Genetic algorithms are based on simplified evolutionary processes that search for optimal results.

Review of different modelling methods in health care can be found in (Stahl, 2008). In GRiST, the inference is performed by using the Galatean model. This research has tried to use other statistical and machine learning based techniques to predict risk and compared them with the clinicians given risk.

2.5 Scope and Challenges of CDSS

The NHS and Community Care Act (1990) had moved the emphasis from care in institutions to care in the community. As a result, scope and need for a reliable and user-friendly decision support system has increased (V. K. Sharma et al., 2010, p. 497). Research shows that clinical decision support systems improve both patient outcomes, as well as the cost of care (Berner & La Lande, 2007). Better decisions due to use of

CDSSs are likely to lead to higher patient safety with better treatment quality, less adverse events and reduced costs (Beeler, Bates, & Hug, 2014).

The Mental Health Taskforce published its report in February 2016, which highlighted the need to improve access to high-quality care for all. “The introduction of the access and waiting time standard for early intervention in psychosis (EIP) services and improving access to psychological therapy (IAPT) services heralded the start of a new approach to deliver this improved access and embed standards akin to those for physical health” (NHS England, 2016, p.5).

Many individuals may underreport stigmatized behaviours in person in an attempt to avoid shame or embarrassment, but web-based screening methods may encourage them to seek help (Michaels, Chu, Silva, Schulman, & Joiner, 2015). They stated that an online methodology could increase the ease by which participants at high or imminent risk of suicide access the service. According to them, the field of psychology is progressively using the internet for interventions and assessment, and it allows reaching large segments of the population easily.

A study by Kim et al. (2012) found that two information quality factors (information reliability and decision supporting capability) and one supporting factor (departmental support) significantly influence user satisfaction. They also found that the ease of use was also a significant factor. A review conducted on the performance of CDSS in literature and it is reported that electronic health records (EHRs) without clinical decision support (CDS) should not be expected to improve quality (Moore & Loper, 2011).

Despite increasing emphasis on CDSS in improving care and reducing costs, evidence supporting its widespread use is limited (Bright et al., 2012). A review of CDSS for the health industry, its success and related risk conducted by Al-gamdi (2014) reports the following factors for non-acceptance of CDSS:

- The system is complex, as a huge knowledge base needs to be searched in order to reach a decision.

- Time-consuming, for doctors and nurses as they are usually busy dealing with the complex diagnosis and do not have enough time to try new systems.

- Lacking decision accuracy, very few systems have reached the high level of accuracy that matched the diagnostic performance of medical professionals.

- Lacking system usability, most early-developed systems were not user-friendly and required a lot of training.

Analysis of 70 randomised controlled trials done by Kawamoto et al. (2005) identified four features strongly associated with a decision support system's ability to improve clinical practice:

a) Decision support provided automatically as part of a clinical workflow,
b) Decision support delivered at the time and place of decision making,
c) Actionable recommendations provided, and
d) Computer based.

GRiST is a web-based, easily accessible system and based on a psychological model. However, it currently lacks background analysis and alert functionality, which is very important according to the above review. This research has endeavoured to add more notification and recommendation functionality to the GRiST system to make it more effective for its purpose. We would like to achieve this by text processing, automatic risk prediction, analysing GRiST node relationships and improving the reliability of the risk assessment. The following sections review existing literature, their shortcomings and link them to our research techniques.

2.6 **Clinical Text Processing**

Unstructured clinical texts may contain a rich amount of patient information but are not immediately accessible to any clinical application systems that require structured input (Y. Wu et al., 2012). Automatic extraction of concepts from the text is a prerequisite for

many NLP applications (Aronson, 2006). Languages are extremely expressive and often there are various different ways to describe the same medical concept (H. Chen, Fuller, Friedman, & Hersh, 2005). To use comments left in the GRiST system for further processing we have considered first extracting concepts from them. In this section, first we have reviewed generic keyphrase extraction methods, then discussed symptom extraction and phrase filtering methods in the existing literature.

2.6.1 Key Phrase Extraction

There are many approaches by which keyphrase extraction can be carried out, such as supervised and unsupervised machine learning, statistical methods, rule-based methods, domain specific methods and linguistic ones (Siddiqi & Sharan, 2015). The following paragraphs describe some of the well known systems and their methods.

The KEA (Keyphrases Extraction Algorithm) calculates feature values using the lexical methods for each candidate phrase and then uses a machine-learning algorithm to predict which of the candidates are good keyphrases. It builds a prediction model using training documents with known keyphrases, and then uses the model to find keyphrases in new documents (Witten, Paynter, Frank, Gutwin, & Nevill-Manning, 1999).

H. Shi, Zhou, Qian, & Li (2009) used Semantic Role Labelling (SRL) based on the dependency trees with multi-features to extract key phrases. Rapid Automatic Keyword Extraction (RAKE) is an unsupervised, domain-independent, and language-independent method for extracting keywords from individual documents (Rose, Engel, Cramer, & Cowley, 2010). In this method, candidate phrases are generated by using stop words and phrase delimiters. RAKE ranks them by the sum of the scores for each of its words. The words are scored based on their frequency and co-occurrence.

Parameswaran (2010) used k-gram to extract concepts. He defined k-gram as a concise entity, which does not contain any extraneous words so that excluding them would identify the same entity. Sharma, Swaminathan and Yang (2010) developed an

algorithm that first identifies and extracts the main verb(s) and using the main verb(s), it then extracts entities of a relationship with the main verb.

Kp-miner uses heuristic methods such as term frequency and position of the term to extract key phrases. They assert that a phrase is never separated by punctuation marks or stop words and a total of 187 common stop words (the, then, in, above, etc.) were used in the candidate key phrase extraction step (El-Beltagy and Rafea, 2009).

The TextRank is based on an unsupervised method. It first tokenises the sentence and then creates a graph of keywords based on their co-occurrence in a certain window. The related keywords are then ranked to obtain the top-ranked keywords (Mihalcea and Tarau, 2004).

The DIKpE system works on three main steps: (i) extract candidate phrases from the document (ii) calculate feature values for candidates (iii) compute a score for each candidate phrase from its feature values and rank them in order. The highest ranked phrases are considered as key phrases (Pudota, Dattolo, Baruzzo and Tasso, 2010).

DegExt is an unsupervised, graph-based, cross-lingual key phrase extractor. DegExt creates a graph representation of words in a document and unlike the traditional vector-space model it takes into account some structural features of the document (Litvak, Last and Aizenman, 2011).

J. Chen, Yan, Zhang, Yang, & Chen (2006) studied topic phrase extraction through Latent Semantic Analysis (LSA). Wang, Mu and Fang (2008) improved Automatic Key phrase Extraction by using semantic information from WordNet. Tomokiyo and Hurst (2003) extracted key phrases by using various statistical measures of their patterns.

A review of different key phrase extraction techniques can be found in (Siddiqi & Sharan, 2015). We would like to extract phrases that are relevant to medical concepts or symptoms. Hence, the following section describes some of the symptom extraction methods found in literature.

2.6.2 Symptom Extraction

There has been a lot of research in biomedical natural language processing over the last two decades. Information retrieval is a significant issue in the medical and healthcare domains where the accuracy of the retrieved information and obtaining it in a time critical situation is extremely important (Patrick, 2009, p. 1). In this section, key phrase extraction research that has analysed biomedical texts is reviewed.

Love, Cai, & Karlson (2011) used text mining techniques to extract data from electronic clinical notes for psoriatic arthritis (PsA) and showed a positive predictive value (PPV) of 93% (89%-96%) when validated with new data. They used simple NLP techniques to search for terms within the clinical notes. The Intelligent Clinical Notes System (ICNS) is a novel system to extract concepts from clinical notes which were written as free text (Patrick, 2009). It tokenises text and matches the token with the SNOMED-CT and other gazetteer.

Gerbier et al. (2011) demonstrated the feasibility of developing an automated method for extracting and encoding medical concepts from emergency departments (ED) narrative reports with an overall recall of 85.8%. They stated that the most frequent cause of failure was non-recognition of the term 9.7% of the time. Overall precision was 79.1%.

MediClass compares phrases against normalized string representations of UMLS Meta-thesaurus concepts to locate concept matches. Concept matches are scored for ‘‘goodness’’ according to the number of changes that need to match the original text segment (Hazlehurst, Frost, Sittig, & Stevens, 2005).

The MedIE (MEDical Information Extraction) system extracts and mines terms from clinical records by three major steps. The first is Ontology-based Term Extraction. The second, also the major one, is Graph-based Relation Extraction. The last is Decision Tree Based Text Classification (X. Zhou, Han, Chankai, Prestrud, & Brooks, 2006) .

“A SympGraph has symptoms as nodes and co-occurrence relations between symptoms as edges, and can be constructed automatically through extracting symptoms over sequences of clinical notes for a large number of patients” (Sondhi, Sun, Tong and

Zhai, 2012, p. 1). They described a symptom expansion method that expands a given set of symptoms to other related symptoms by analysing the underlying SympGraph structure.

McCart et al. (2012) explored multiple approaches; combining regular expression-based rules, statistical text mining (STM), and an approach that applies weights to text while accounting for multiple labels to analyse suicide notes. They achieved a micro-averaged F1 score of 0.5023, slightly above the mean (0.4875) from the other 26 teams who competed.

MedLEE is a web-based system, which uses an automated system for acquisition and discovery of medical knowledge embedded in clinical narrative reports. It uses statistical methods and a random sample of disease-symptom associations; it indicates an overall recall of 90% and a precision of 92% (Xiaoyan Wang, Chused, Elhadad, Friedman, & Markatou, 2008).

Griffith et al., (2012) have developed a rule-based algorithm and evaluated a natural language processing (NLP) system for infectious symptom detection from clinical narratives. They trained the system with related keywords and SNOMED-CT concepts.

Koeling, Tate and Carroll (2011) used the UK General Practice Research Database (GPRD), which contains coded data supplemented by free text (physicians' notes and letters). They found that the system could estimate a 40% higher number of symptoms, when coded information was enhanced by manually tagged free text.

A system was developed by Gorrell et al. (2016) to find the cases of first episode of psychosis using machine learning techniques that achieved an area under curve (AUC) of 0.85, enabling 95% of relevant cases to be identified, whilst halving the work required in manually reviewing cases. They used manually annotated data and machine learning.

A hybrid system developed with a support vector machine (SVM) learning algorithm and rule-based text matching, using the Generalised Architecture for Text Engineering (GATE) software package, has extracted negative symptoms from the clinical narratives of patients with schizophrenia. A substantial proportion (41%) of the sample that was analysed by this system had at least two negative symptoms (Patel et al., 2015).

An important part of the phrase or symptom extraction is to filter irrelevant phrases and only keep the domain relevant phrases. Before describing our intended approach, some of the filtering methods found in literatures are described in the following section.

2.6.3 Phrase Filtering Methods

This is an active area of research and there are many phrase ranking algorithms available. In fact, all the phrase extraction methods described earlier have their own filtering logic. All phrase extraction methods in one way or another include a phrase filtering procedure. Literatures were reviewed to find a suitable algorithm for the filtering task. Pudota, et al. (2010) proposed a frequency based algorithm. They have used the first and the last occurrence of a phrase in a document as a feature. As our data comes from the different GRiST ontology nodes, so such features are not applicable in our case.

J. Chen et al. (2006) described a latent semantic analysis based method. Bleik, Xiong, Wang, & Song (2010) described a method using a concept graph. Wan & Xiao (2008) described a method extracting phrases from a single document by constructing a small set of neighbouring documents. These ideas are considered in the developing of the proposed ECM method, which is described in Chapter 4.

Zhao et al. (2011) described a probability and PageRank based method to extract phrases from Twitter. The GRiST data is different from the Twitter data. Kumar & Srinathan (2008) used n-gram to extract candidate phrases and then filtered them. For filtering, they have used sentence position in the documents, which is not applicable to our data.

Xin Jiang et al. (2009) described a method of extracting phrases by comparing them with a seed phrase list. They used co-occurrence frequency as the basis of new concept selection. The KP-miner system does not need training data it uses heuristics to extract phrases (El-Beltagy & Rafea, 2009). It is highly dependent on three main steps: candidate key phrase selection, candidate key phrase weight calculation and final key

phrase refinement. A review of different filtering methods can be found in (Hasan & Ng, 2010) and (Siddiqi & Sharan, 2015).

From the above review of the key-phrase extraction methods, we see that most of the automatic methods used heuristics or statistical measures for relevancy filtering or used a manually created word list. A generic phrase extraction method may produce domain irrelevant phrases. On the other hand, domain specific supervised methods need human-annotated training data and may only perform well in a specific context (Pudota et al., 2010). These type of methods require annotation and training for every new domain.

We proposed a novel method whereby we extract phrases and filtered them by using word embedding based semantic filtering to extract domain relevant phrases. The word2vec model developed by Mikolov et al. (2013) represents a word semantically and conveys semantic meanings. The key phrases can be first extracted by a generic method and then a semantic filtering method would keep only the domain relevant phrases. For the i2b2 dataset, our approach provided a better result than the RAKE or OpenNIP approaches. The proposed method is described in Chapter 4 of this report.

2.7 **Suicide Risk Prediction**

There are many general purpose text-based classification tools available. As described below, a few researchers have also claimed good results for classifying mental health problems from clinical notes. In this research, we have used many existing toolsets and methods to classify suicide risk by using clinical notes. The following paragraphs describe some of the relevant research papers.

While electronic health records are invaluable for medical research, much of the information is noted in text form rather than the coded form (Shah, Martinez, & Hemingway, 2012). For example, causes of death and test results recorded in the UK General Practice Research Database (GPRD), are sometimes only recorded in free text (Shah et al., 2012). Free text is difficult to use in research and it often requires manual

reviewing. An automatic matching system is proposed by Shah et al. (2012) to find the cause of death from text data by using a look-up table.

A linguistics-driven prediction model was developed by Poulin et al. (2014) to estimate the risk of suicide from clinical notes. The models were built with the unstructured clinical notes collected from a national sample of U.S. Veterans Administration (VA) medical records. They have created three matched cohorts: veterans who committed suicide, veterans who used mental health services and did not commit suicide, and veterans who did not use mental health services and did not commit suicide during the observation period (n=70 in each group). From the clinical notes, they have manually created a list of single word and multi-word phrases, and constructed prediction models based on a genetic programming framework. According to them, the resulting inference accuracy was consistently 65% or more. They concluded that computerised text analytics could be applied to unstructured medical records to estimate the risk of suicide. The resulting system could screen people at primary care level and continuously evaluate suicide among psychiatric patients (Poulin et al., 2014).

Automatic detection of suicidality in Twitter was investigated by O'Dea et al. (2015). This study examined whether machine learning could replicate the judgement of human coders on expression of suicide in Twitter posts. They have collected 14,701 suicide-related tweets and 2000 of the tweets classified by human coders. Overall, 14% of the tweets were classified as 'strongly concerning', with the majority coded as 'possibly concerning' (56%) and the remainder (29%) considered 'safe to ignore'. The inter-human coders agreement was 76% (average κ= 0.55). Machine learning processes were then applied to assess whether a 'strongly concerning' tweet could be identified automatically. The computer correctly identified 80% of 'strongly concerning' tweets; however, predictability decreases as the data size increased (O'Dea et al., 2015).

Yang, Spasic, Keane, & Nenadic (2012) present a system developed for the i2b2 obesity challenge to identify obesity status and 15 related co-morbidities in patients from their clinical discharge summaries. According to them, the challenge consisted of two tasks, textual and intuitive. The textual task was to identify the explicit references to the diseases and the intuitive task was to find disease status when the evidence was not explicitly present (Yang, Spasic, et al., 2012).

Pestian et al. (2008) hypothesised that the machine learning algorithm (MLA) can classify completer and simulated suicide notes as effectively as any mental health professional (MHP). Five MHPs classified 66 simulated or completer notes and later machine learning was used on the same data. The result shows that MHPs were accurate 71% of the time and using the sequential minimisation optimisation algorithm (SMO) MLAs were accurate 78% of the time. They concluded that there was no significant difference between the MLA and MPH classifiers and an evidence-based suicide predictor for emergency departments can be developed (Pestian et al., 2008).

In the US suicide is the tenth leading cause of death and considering the significance the Informatics for Integrating Biology and the Bedside (i2b2) Natural Language Processing (NLP) shared task competition (track two) was focused on suicide (McCart et al., 2012). The challenge concentrated on sentiment analysis, predicting the presence or absence of 15 emotions (labels) simultaneously in a collection of suicide notes spanning over 70 years. The author's team found multiple approaches including regular expression-based rules, statistical text mining (STM), and text weighting approach (McCart et al., 2012).

Existing methods for the event trigger identification typically rely on annotated training data where the event trigger words are labelled with their corresponding event types (D. Zhou, Zhong, & He, 2014). The framework proposed by the authors, learns biomedical knowledge from a large text corpus built from Medline and embeds it into word features using neural language modelling.

Another project submitted to the (i2b2) Track 2 Shared Task for sentiment analysis in suicide notes has used hybrid methods. The proposed hybrid model incorporates a number of natural language processing techniques, including lexicon-based keyword spotting, CRF based emotion cue identification, and machine learning-based emotion classification (Yang, Yang, Alistair Willis, Anne de Roeck, & Bashar Nuseibeh, 2012).

Metaphorical analysis has been used by Neuman, Cohen, Assaf, & Kedma (2012) to infer the existence of depression in the text, given the variety of linguistic means one may use to express it. They created a list of metaphors to detect depression in free text.

None of the above researchers has used word embedding to predict suicide risk. We have assumed that we can use all of the related text of a patient as a document. Then create a document vector by using word-embedding techniques and then use the vectors to classify patients. We have used a larger dataset and multiple techniques such as raw text, SNOMED-CT code in the text and word2vectors of the text. Being able to classify risk levels using text data could help improve the GRiST system. Many experiments were carried out on the GRiST dataset to predict suicide risk using both text and other methods.

2.8 Frequent Pattern Mining

Classification algorithms often do not predict all the classes with equal accuracy. With GRiST dataset, we have found that the prediction of high suicide risk was not accurate. It is important to predict higher risk categories more accurately. To identify high-risk patients, we have applied the node relationship and frequent pattern mining approach.

Pattern mining may allow us to notify clinicians as soon as the risk related pattern is identified. The itemset or pattern that can predict risk may rarely occur. To address this issue and improve the result we propose a method that looks for multiple rare patterns. From the experimental data, we have shown that our approach improves the precision of the classification. First, we have used chi-square analysis and then frequent itemset mining.

2.8.1 Chi-square Analysis

The Chi-square (X2) test is a nonparametric statistical test to determine if the two or more classifications of the samples are independent or not (Zibran, 2015). The chi-square statistic may be used to test the hypothesis of no association between two or more groups, populations, or criteria. Knowledge of associations between biomedical entities, such as disease-symptoms, is critical for many automated biomedical applications (Xiaoyan Wang et al., 2008).

Finding relationships among medical symptoms, disease and treatment is an active area of research. The Clinical E-Science Framework (CLEF) project is used for identification of relationships between clinically important entities in the text by Abdel-moneim, Abdel-Aziz, & Hassan (2013). They have identified entities that relate to the custom define classes.

A machine learning based system for relation extraction implemented by Roberts, Gaizauskas, Hepple, & Guo (2008), using support vector machines, was trained and tested on corpus of oncology narratives that was manually annotated for clinically important relationships. Over a class of seven relation types, the system achieved an average F1 score of 72%, only slightly behind the human inter-annotator agreement on the same task. There has been growing interest within scientific communities to use text mining tools to find knowledge such as protein-protein interactions (D. Zhou & He, 2008).

Yang, Yang, et al. (2012) supplemented manually created list of emotional terms by a list of terms that were selected from the annotated emotion instances and were identified as significant by Pearson's chi-square from suicide notes. A list of available senses can be created using Wordnet for given documents. A new concept term was linked with the documents using the chi-square statistic. The word sense with the highest chi-square score was the chosen sense for that concept candidate (K. Liu, Hogan, & Crowley, 2011).

Chi-square Automatic Interaction Detection (CHAID), and association rules were used to identify factors affecting the sentiments of adolescent depression (Jung, Park, & Song, 2017). Byeon (2017) developed a depression prediction model for female students from multicultural families by using a decision tree model based on the CHAID algorithm. Outcome variables were classified as presence of depression. Explanatory variables included sex, residing area, experience of career counselling, experience of social discrimination, experience of Korean language education, experience of using a multicultural family support centre, Korean reading, Korean speaking, Korean writing, Korean listening (Byeon, 2017).

When each variable has a little marginal effect, it is difficult to discover predictive variables. The interaction between predictive variables may be used to predict the

outcome (Xia Jiang, Jao, & Neapolitan, 2015). First, they identified candidate interactions by determining whether together variables provide more information than they do separately.

It could be useful to find out how GRiST nodes interact with each other. Most of the previous research worked on the disease-symptom interactions or protein-protein interactions. We are trying to analyse the relationships among symptoms (GRiST nodes) and assess their impact on the patients' overall suicide risk level. This might help us to discover relationship patterns that assist in making the GRiST system more effective and interactive.

2.8.2 Frequent Itemset Mining

The GRiST node relationships may be learned by association rule mining. Recently, pattern mining techniques are being adopted as a core part of many bioinformatics solutions and frequent itemset mining has been used to identify elements such as disease and symptoms that frequently co-occur (Naulaerts et al., 2015). This is a non-trivial problem and a number of algorithms have been developed. According to Naulaerts et al. (2015) frequent itemset mining techniques can capture the characteristics of complex data and succinctly summarise it and these techniques have demonstrated its usefulness in biomedical data analysis.

A mobile interface was created by Huang, Huang, Chen, Liu, & Huang (2012) to find associations among users' responded questionnaire and their negative emotion. They have used FP-tree (Frequent Patterns tree) and FP-growth (Frequent Patterns growth) algorithms to discover the interesting association rules from users' negative emotions. The have used support of 2% with FP-growth algorithm.

Lakshmi & Kumar (2014) have described a new method of uncovering valid association rules from medical transcripts. The extracted rules describe the association among diseases, symptoms of a particular disease, medications used for a disease and the most prominent age group for a disease. They have used NLP (Natural Language Processing) tools in combination with data mining algorithms (Apriori algorithm and FP-

Growth algorithm) for the extraction of rules. They have claimed that the method was helpful in finding the diseases that most likely co-occur with diabetes and also the medications used in treating diabetes.

FP-growth association rule mining algorithm was used for detection of diabetes at an early stage by Rane & Rao (2013). Apriori association rule mining was applied to case diagnosis of the breast cancer in the hospital to find out the association of factors from volumes of case recordings (W. Zhang, Ma, & Yao, 2014).

The development of mental illness can be related to a variety of psychological factors (Lacković et al., 2014). They have used association rules to predict mental illness based on scale scores of five psychological factors, including family functioning, social support, depressive symptoms, perceived empathic self-efficacy, and anxiety disorder. They have used support 1.5% since mental disorders are relatively rare in the healthy population.

A review of the different pattern mining techniques can be found in (Satpute, 2014). A general survey and comparison of the algorithms for association rule mining can be found in (Hipp, Güntzer, & Nakhaeizadeh, 2000). An overview of the various algorithms and illustrations of their use in several real-life bioinformatics application domains is provided in (Naulaerts et al., 2015) and (K. P. Kumar, 2013).

The above review shows that because the risk of disease is a rare event, many researchers have used a very low support value to extract association rules. Use of low support may affect the accuracy of the prediction. We have proposed a new method by which we can apply multiple rules and achieve greater accuracy in predicting high suicide risk. We have used the FP-growth algorithm to predict higher suicide risk in patients. We used symptoms association to predict suicide risk rather than only analysing inter symptoms relationships.

2.9 **Reliability of Risk Assessment**

The use of CDSS systems is increasing and their use is aimed at improving patient safety. There is significant evidence that CDSS can positively impact healthcare

providers' performance with drug ordering and preventive care reminder systems (Jaspers, Smeulers, Vermeulen, & Peute, 2011). Evidence that CDSS significantly impacted processes of care was found in 108 out of 143 unique studies (Jia, Zhang, Chen, Zhao, & Zhang, 2016). Research has suggested that users often over-rely on system suggestions, even if the suggestions are wrong and providing explanations could potentially mitigate misplaced trust in the system and over reliance (Bussone, Stumpf, & O'Sullivan, 2015).

We may use information gain to improve accuracy of risk judgement and provide better feedback to clinicians. The presence of different symptoms may not be equally important to diagnose a disease. This could be quantified by measuring the information gain of a symptom. Information gain has been used to improve prediction/ classification accuracy, some of the previous researches are described below.

A support vector machine (SVM) and information gain based classification framework for Diabetic Retinopathy Images has been described by Dharani, Menaka, & Vinodhini (2014). Their experimental result shows that the Information Ranker-PART was faster than the SVM but the SVM had a lower mean square error.

Ambert & Cohen (2012) described an algorithm based on information gain. They have used kNN and modified it to select features based on information gain of the features. The new algorithm was called kIGNN. They concluded that the performance of kIGNN was better than the kNN baseline, and it was mainly due to the use of information gain in kIGNN as that was the only significant difference between the two methods. They noted that the choice of classification algorithms had the most influential contribution to the performance of the systems. SVM light performed better with a smaller training set (Ambert & Cohen, 2012).

Maucort-Boulch, Roy, & Stare (2014) discussed the measures of fitness of a regression model. If we are interested to know how much of the variation in the outcome variable is explained by the model, then in the case of simple linear regression, we can use the well-known R-square measure. They have argued that for nonlinear relationships the information gain might be a better choice.

A unifying measuring criterion was proposed by Shtatland & Barton (1997), along with the use of other criteria such as R-Square, deviance, log-likelihood and so on. According to them, the proposed information difference statistic is better for the following reasons:

a) It is common for all the types of regression analysis;
b) It is easy to interpret in terms of information gain/loss (in bits);
c) It has a very convenient property of additivity that allows system users to evaluate the contribution of an individual feature in terms of information.

In machine learning and classification, high dimensional data can be challenging to handle, the so-called 'Curse of Dimensionality'. One of the interesting ways to handle this could be to look for interaction between data attributes and use that in machine learning (Jakulin, 2005). Xia Jiang, Jao, & Neapolitan (2015) addressed this problem using information gain and Bayesian network scoring. First, they identified candidate interactions by determining whether together variables provide more information than they do separately. Then they used Bayesian network scoring to find out if a candidate interaction really is a likely model.

Xia Jiang, Jao, & Neapolitan (2015) presented MBS-IGain, a method for identifying interactive effects in high dimensional datasets. Based on their experiments, MBS-IGain was highly effective at doing this, substantially exceeding other methods. Orimaye, Wong, & Golden (2014) have used information gain to choose features for classification of Alzheimer's disease from textual narratives.

The above review describes the application of information gain to improve classification accuracy. We believe we can use the same concept to validate the reliability of the clinical judgement. The proposed approach of validating risk assessment is novel. If a clinician asks all the relevant questions and thus gains more information about a patient, then it could be assumed that the clinician is more likely to make an accurate risk judgement.

When clinical judgements vary significantly from calculated risk, we may use information gain to explain the difference. We have used the sum of total of the answered questions each weighted by their information gain. We have hypothesised that if more information was collected the clinical judgement would be more accurate. To overcome the problem

of the interrelationship among GRiST nodes, we later used total “Relative weights” instead of information gain. Relative weight is the amount of variance explained by an attribute and it compensates for the attributes inter correlation. The methods and results are explained in Chapter 8.

2.10 **Summary**

In this chapter, I have explained the background and have highlighted the importance of this research. The existing literature has been reviewed, and an introduction is given on how our research extends the past research. We envisioned to exploit the presence of symptoms in clinical notes to predict suicide risk, hence concept phrase extraction methods have been reviewed. Particularly we were interested to extract relevant concept phrases automatically from clinical notes.

We also aspired to explore GRiST nodes inter relationships and its impact on suicide risk to alert clinicians at the early stage of the assessment. Information gain and related technologies have been reviewed as a measure to quantify the validity of a risk assessment. The GRiST system was used as a test case CDSS. The next chapter discusses the GRiST system in detail.

3 The GRiST CDSS

3.1 Introduction

The Mental Health Action Plan 2013-2020 of the World Health Organisation (WHO) recommends “the development of comprehensive community-based mental health and social care services; the integration of mental health care and treatment into general hospitals and primary care; continuity of care between different providers and levels of the health system; effective collaboration between formal and informal care providers; and the promotion of self-care, for instance, through the use of electronic and mobile health technologies” (World Health Organization, 2013, p. 14). An easily accessible mental health assessment tool could facilitate to achieve these objectives and could save money and resources.

GRiST is a web-based assessment tool and it is accessible from any place at any time. The use of web-based assessments and intervention systems are increasing in psychology (Michaels et al., 2015). They noted a few advantages of web-based systems, which include increased validity, feasibility, efficiency, as well as improvements in data collection and management.

The Galatean Risk and Safety Tool (GRiST) is a mental health assessment tool developed by Aston University and Warwick University. It is based on the Galatean risk assessment model proposed by Buckingham (2002). It is a Clinical Decision Support System (CDSS) used to evaluate multiple types of mental health related risks including suicide, self-harm, harm to others, self-neglect, and vulnerability. GRiST differs from alternative risk assessment tools by its use of a psychological model of classification. A detailed description of GRiST is provided, as it is the primary source of all the data used in this research.

According to Buckingham & Adams (2011) the goal of GRiST is to “provide universal access to validated expert advice on risk judgements that can be clearly understood by people without a specialist mental-health background and be flexibly presented according to end-user requirements” (p. 1). It is currently being used by NHS secondary

mental health trusts, private hospitals, charities, primary care IAPT services, and for self-assessment in the community. “The aim is to facilitate risk communication across the care pathway and give patients more involvement in monitoring and managing risks” (Buckingham & Adams, 2011, p. 1).

3.1.1 Commencement of the GRiST Project

“GRiST project was funded by an NHS New and Emerging Applications of Technology grant, and Multi-Research Ethics Committee clearance was obtained. The research began with 46 practitioners who were from multiple disciplines and backgrounds which ensured to provide multiple perspectives and experiences of risk assessment, encompassing academic and research areas, as well as clinical practice. Most of the 46 participants who were interviewed came from psychiatric nursing (21) and psychiatry (13), but there were also some social workers, general practitioners, and psychologists” (Buckingham, Ahmed, & Adams, 2007, p. 1).

According to Buckingham et al. (2007) people were recruited on a continuous basis throughout the GRiST project and many of them were involved in web tasks and focus groups. The final panel membership consisted of over 100 clinicians and service users. As this research relies on data provided by GRIST, detailed knowledge of its development and structure is required.

The following is the summary description of GRiST development process adopted from (Buckingham et al., 2007).

Step 1: Interviews were transcripted in the documents and the concepts were extracted from them using thematic analysis.

Step 2: A mind map was built from each of the interviews using a mind map coding template. The process was evolved as the interviews were analysed.

Step 3: All the mind maps were combined, which represented the comprehensive risk-assessment knowledge from interviews with 46 multidisciplinary mental-health experts.

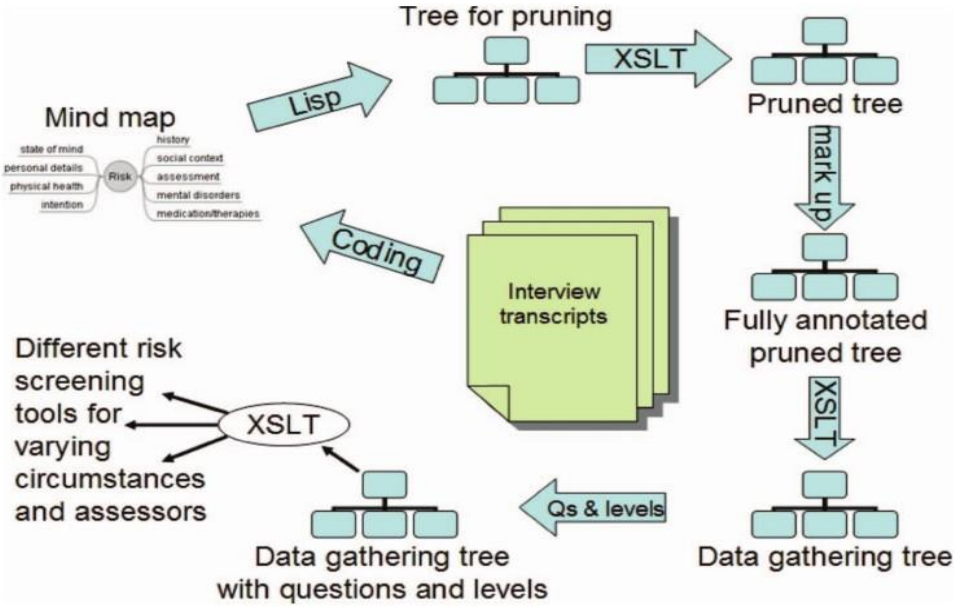

Figure 1 GRiST ontology development process reproduced with permission from (Buckingham et al., 2007)

Initially, the combined structure had 7210 nodes of which only 1432 were unique. Of the unique nodes 477 were concepts (i.e. have child nodes) and remaining 962 were leaf nodes. Some of the nodes were generic and they apply to all types of risk. The mind map was saved in XML format, which enabled easy accessibility to the information by XSLT queries.

Step 4: Clinicians prefer tools that do not involve asking too many questions. Hence, the big tree was pruned with the feedback from focus groups. A custom Flash based tool was developed to mark the node for pruning visually. The pruned tree had a total of 394 nodes, of which there were 124 unique concepts and 228 unique leaves.

Step 5: The next step was to combine all these nodes in an XML file in accordance with the Galatean Model of classification. A partial schema of the XML is shown via an example node later in this chapter. This XML tree used as an underlying knowledge base for GRiST CDSS, which allows to quantify mental health risk.

“GRiST is accessed through simple web-based browsers or mobile devices and its advice help determine whether the potential risk associated with a person justifies a more detailed assessment by a specialist clinician. GRiST’s potential for flexible interfaces means the specialist clinician, too, will be able to use it for conducting the assessment, thereby providing a seamless transmission of risk information that transcends disciplines and services” (Buckingham et al., 2007, p. 79).

3.1.2 GRiST Innovation

GRiST is not only an easy to use web-based system, it also has an underlying validated psychological model. According to Buckingham and Adams (2008) GRiST is a CDS system, which is based on the human psychological model of decision making. They argue that GRiST's psychological model of classification distinguishes it from alternative risk-assessment tools.

“GRiST uses the Galatean model of decision making which mimics how people use cues to make decisions. It is based on the premise that the probability of different decision outcomes compete with each other for influence on the final decision. GRiST explicitly captures structured clinical judgement and links it to sophisticated probabilistic and statistical analyses of the patient database” (Buckingham and Adams, 2011, p. 2).

While there are plenty of tools that take evidence in isolation (e.g., history of previous attempts increases suicide risk), Buckingham, Kearns, & Brockie (2004) argued that none of them has succeeded in identifying how combinations or patterns of cues can be integrated to give a single, accurate risk prediction.

Considering the data for this study came from the GRiST and they are collected based on the Galatean model, a detailed description of the Galatean model is given in the next section.

3.2 The Galatean Model

The underlying classification model of GRiST is a called the Galatean Model. The Galatean model is grounded in prototype theory. According to Hampton (2006) Prototype can be considered as a generic abstract concept that represents the similarity of the category members, and the differences from non-members. It is the centre of the cluster of similar objects. The centre of the cluster is well established, but the boundary between one category and another may be subject to vagueness (Hampton, 2006).

“The galatean model is a type of prototype model but instead of representing the average class member, its prototype encapsulates the hypothetical ‘perfect’ member, the one with the highest probability of membership (the name ‘galatea’ comes from Pygmalion’s perfect woman)” (Buckingham, 2002, p. 240).

The base-rate fallacy is a tendency of human beings ignoring available statistical data in favour of specific data to make a probability judgement, rather than integrating the two. This tendency has significant implications for understanding judgement phenomena in many clinical, legal, and social settings (Bar-Hillel, 1980). The Galatean Model considers this phenomenon and it was designed to encapsulate the real-world expert’s decision making process. A detailed explanation of the underpinning of the Galatean Model can be found in (Buckingham, 2002).

A patient can have attributes such as age, ethnicity, hair style, etc. and clinical symptoms such as blood pressure, pulse, temperature. A specific value of an attribute is referred to as a cue. Clinical decisions can be considered as classification tasks where cues are used to classify a patient to a specific category (Buckingham, 2002).

According to Buckingham (2002) the suicide galatea cues maximize P(Category|cue) instead of P(cue|Category). All the categories, such as suicide and no-suicide, are represented by their own galateas, where “a galatea consists of components for each of the attributes relevant to classifying a person into the associated category” (Buckingham, 2002, p. 241).

Uncertainty is represented in the Galatean model as a set membership. For example, if categories are considered as sets then the likelihood of an object to be in any of these sets are given by its degree of membership to that particular set. The measure of this criteria is called the membership grade (MG) which, like probabilities, may vary from 1 (certain membership) to 0 (no membership) (Buckingham et al., 2004).

In the Galatean Model, mental health risk is represented in a hierarchical tree of nodes. The parent concept nodes may have child nodes and so on. Eventually the leaf node represents an individual cue, which can be referred to as datum. The eventual risk contribution of an individual cue depends on its relative influence (RI) compared to the other sibling cues. RI values are pre-assigned by experts and the total risk attributable to the containing galatea is the sum of the RI-weighted MGs (Buckingham et al., 2004).

The data used in this research was originally collected based on this model. GRiST is being delivered to mental-health organisations (including NHS foundation and hospitals) as a cloud-computing service from Aston University with a database of assessments increasing every day. About 1400 new assessments are being completed each month (source: https://www.egrist.org/why-grist).

3.2.1 Risk Calculation Example

An example may explain the GRiST internal risk calculation process better than just a description. This example is taken from the original paper (Buckingham, 2002) and (Hegazy, 2009). The figure below shows a hypothetical assignment of membership grades and the relative influences on the ‘Intention’ concept and how these produce membership grades for a particular patient.

For example, each three cues (seriousness, realism and steps taken) can have values between 0 and 10, with a value of 10 providing the maximum membership and value 0 the minimum membership grade. If the clinician gives realism a value of 7 than the membership grade will be 0.7. Similarly, ‘Steps taken’ generate an MG value of 0.8.

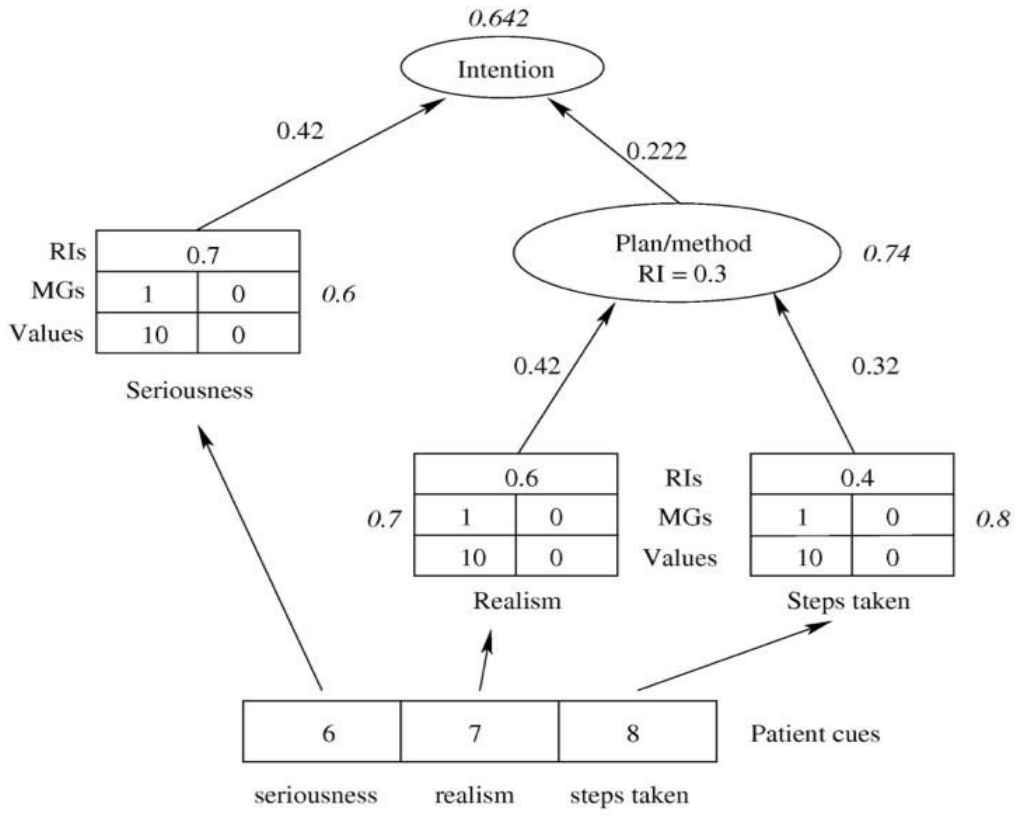

Figure 2 Example of risk calculation, reproduced with permission from (Buckingham, 2002)

Now the relative influence (RI) of the 'realism' is 0.6 and for the 'Steps taken' it is 0.4. Relative influences of all siblings are sum to 1. The membership grade of realism (0.7) is multiplied by the relative influence of 0.6 to pass a membership grade of 0.42 up to the plan/method concept. Similarly, Contribution of 'Steps taken' node is 0.32. The total MG of the 'Plan/method' node is the sum of its child's contribution, which is 0.74 in this example (Buckingham, 2002).

In a similar process, we can multiply the MG (0.6) of 'seriousness' node with its RI (0.7) and get its contribution towards Intention is 0.42. The contribution of 'Plan/method' node is given by multiplication of MG (0.74) and RI (0.3), which is 0.222. The final MG value of the Intention node is 0.42+0.222= 0.642. This means the patient has a 64% intension of committing suicide. By changing the membership grade and relative influences, the experts adjust the classification process and align the class membership grades that correspond to their own estimates of risk (Buckingham, 2002).

3.3 The GRiST Ontology

A short description of the structure of the GRiST ontology and the data collection methods is important as this data is used for this research. The following description is summarised from (Ahmed, 2011) and (Buckingham et al., 2004).

GRIST ontology is a structured tree (ST) available in XML file format. On the top label, we have the mental health risk node. This is then divided into five main areas of mental-health risk. These are suicide, self-harm, harm to others, self-neglect, and vulnerability. A schematic representation of the ST is presented in the following figure:

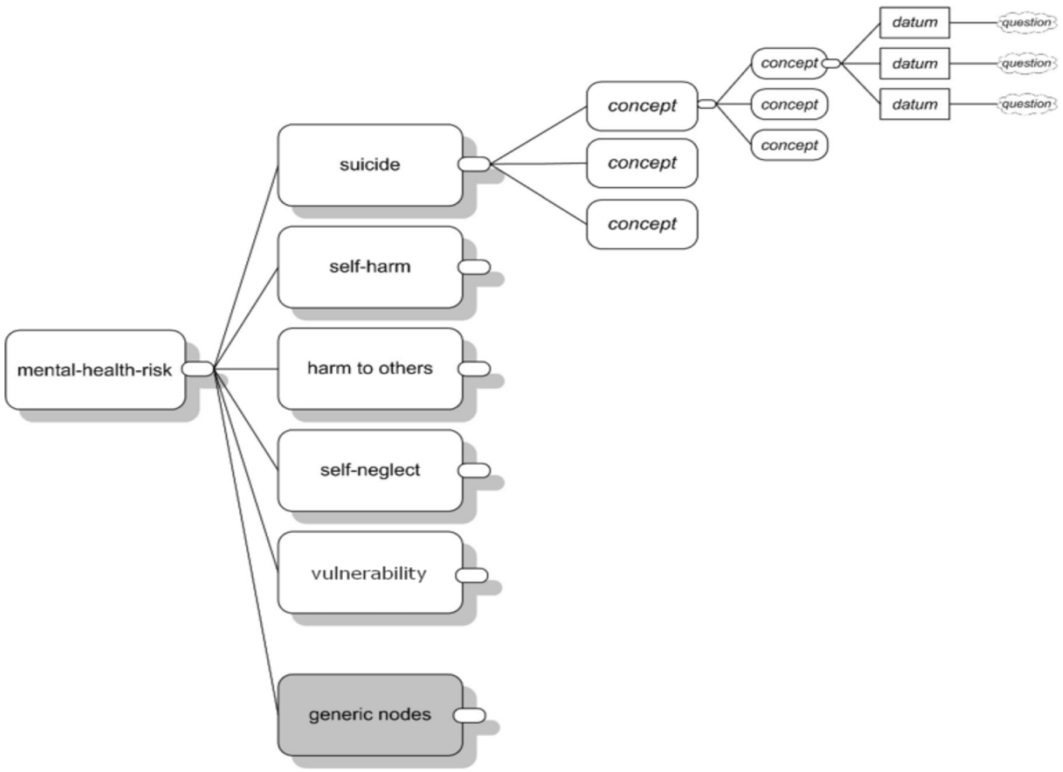

Figure 3 GRiST structure tree (ST) reproduced with permission from (Ahmed, 2011)

Each top-level node is subdivided into its constituent's child nodes. The last nodes in the tree are the individual datum or leaf nodes. Each datum node represents a piece of information that can be collected by clinicians. There is a question attached to the leaf node via an additional question attribute.

At the beginning of the assessment, a full assessment tree is created as per the assessment type and the type of patient. This tree is then used to dynamically create the user interface. GRiST's main user interface is web-based and hosted in a central location. This allows the storing of data in one central location securely and provides access to the service users easily without them needing to install software on their own computers. A sample user interface of GRiST is shown in the screenshot below.

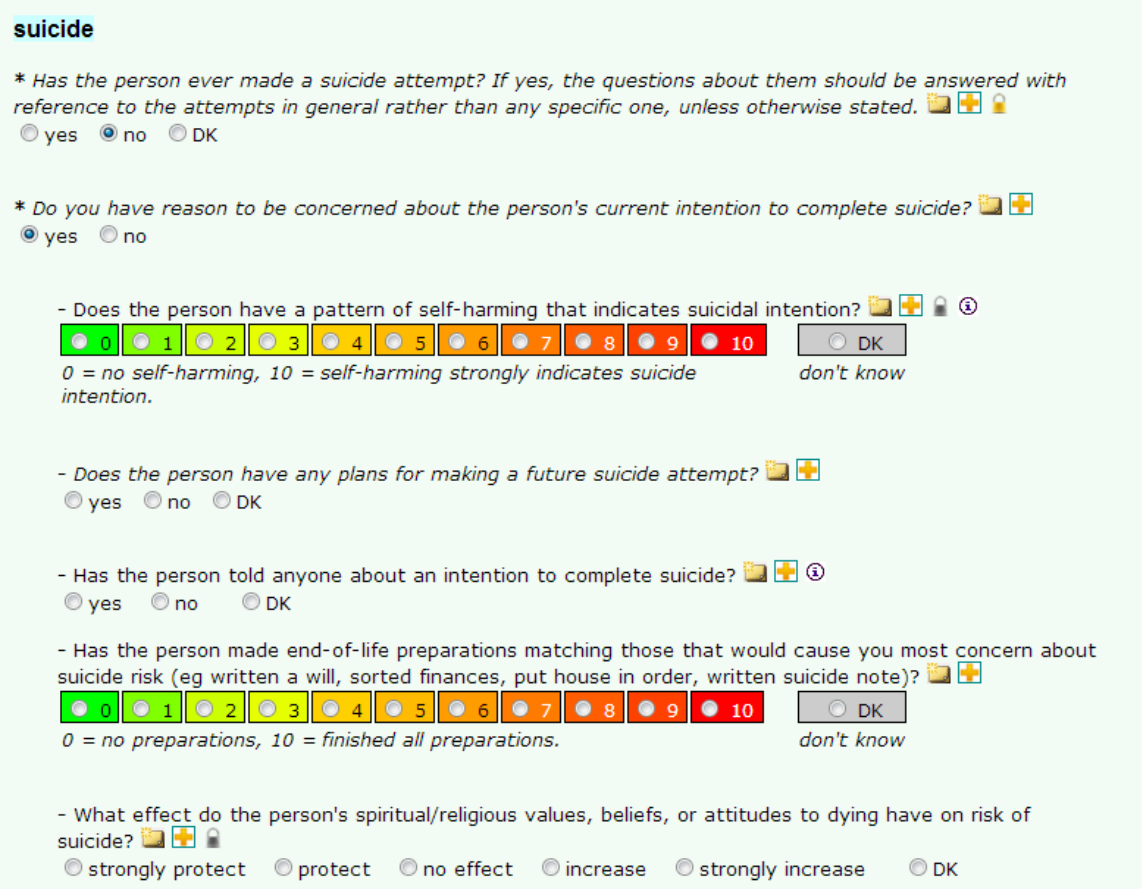

Figure 4 Screenshot of GRiST user interface

If a filter question is answered 'YES' then other child questions under that node are opened up. Each of the nodes has an option to add comments for that particular node. These comments are stored in the database separately. For this research, actual numeric data (mg-value) and its corresponding comments are used.

For a scale data type a shaded list of radio buttons is displayed as shown in the image above. Clinicians can simply click the relevant choice of answers. The plus icon shown in the image is used to input additional comments. Data that are persistent are copied to the next repeat assessment automatically. The GRiST system allows skipping data entry, which causes missing data problems. The input of comments is also optional, which means there are missing comments in the data. The implications of this are described in the relevant experimental sections later in this report.

3.3.1 Generic Concepts

There are some nodes, which are common across the risk type. The GRIST implementation team have put them in a common location. Stub nodes are placed in the tree where necessary to refer to the generic node. This is mainly an internal organisation and the full structure tree (ST) is created for each type of assessment dynamically. Generic nodes for which the RI is fixed are noted by attribute generic-type=”g”. And generic nodes which have different RI for different risk types are marked as generic-type=”gd” (Ahmed, 2011). For my research this was not a concern as the data was collected from the database where all the nodes were present.

Example of an XML node of the GRiST structure tree is shown below.

<node label="suicidal thoughts” code="suic-ideation" question="To what extent do the person's suicidal thoughts/fantasies match those that would give you most concern about suicide risk? " values="scale" value-mg="((0 0) (10 1))" level="1" filter-q="Is the person having suicidal thoughts or fantasies?">
<node label="ability to control suicidal thoughts" code="suic-id-control" question="To what extent does the person lack ability to control suicidal thoughts or fantasies?" values="scale" value-mg="((0 0) (10 1))" "/>
<node label="high risk suicidal thoughts" code="suic-id-hi-risk" question="How much do the ways you are imagining ending your life make it more likely that you will try to do it? " values="scale" value-mg="((0 0) (10 1))" "/>

<node label="frequency of suicidal thoughts" code="suic-id-freq" question="How often do the suicidal thoughts or fantasies occur?" values="nominal" value-mg="((DAILY 1) (WEEKLY 0.5) (MONTHLY 0.2) (LESS-THAN-MONTHLY 0))"/>
<node label="strength, intensity" code="suic-id-strngth" question="How hard is it to get thoughts of suicide out of your head?" values="scale" value-mg="((0 0) (10 1))"/>
</node>

In this research, all the GRiST nodes are saved in the database as a parent child relationship, including the datatype and other meta information. A utility was created to traverse through the nodes quickly for exploration purposes.

3.3.2 Question and Datatype

From the user's perspective, each GRiST node represents a question. These questions can be a generic question, a filter question or additional questions under the filter questions. The answer of the filter question itself is also a data. The most used data type in GRiST is the scale data type. After answering all or partial questions, clinicians finally input their own judgement on the suicide risk. The GRiST nodes can collect any of the following data types:

Table 7 GRiST data type

Type name	Description
Scale	Generally range between 0 to 10
Number	values="integer|real"
Date	the date-year, date-month, date-week, and date-day
Nominal	categorical data
Ordinal	Order of the values

The Galatean model requires membership grade data so each node input data needs to be converted to value-mg. For categorical data value-mg was estimated by focus group discussions (Ahmed, 2011).

All the collected data is saved in the database. Comments and other data are stored in two separate tables. For this research, the data from the database was used. Some filtering was required as provisional 'in progress' assessment data was also saved in the same table with an identifier.

3.4 GRiST Dataset for Experiments

GRiST is provided as a web-based service from Aston University to mental-health organisations and its database of assessments increasing every day. "Currently, the electronic version of GRiST (eGRiST) is being used across a range of clinical services in: Humber NHS Foundation Trust (1600 staff); Cumbria Partnership Trust (1500 staff); Birmingham Children Hospital; Raphael Healthcare, Newark, for their 40-bedded female forensic unit; and the Craegmoor Hospital group."
(Source: https://www.egrist.org/content/some-places-where-grist-being-used).

The data for this research comes from the GRiST project. We have only chosen data which has at least 1KB of comments. There were a total of 46903 instances of assessments, of which 38197 have a suicide risk less than 5 and 8706 have a suicide risk more than or equal to 5. We considered the later group of patients as high-risk patients.

Assessments conducted between 2011 and 2013 (a total of 21203 assessments) were used for training and the rest (25700 assessments) were used for testing purposes. The following table shows the distribution of risk levels in the data.

Table 8 Grist assessment data with risk level

Year/Risk	1	2	3	4	5	6	7	8	9	10	total
2011	802	1037	1148	542	513	216	214	127	49	15	4663
2012	1271	1547	1413	706	577	274	270	165	59	19	6301
2013	2824	2661	1839	995	846	381	368	228	74	23	10239
2014	4007	3228	2191	1208	928	411	395	260	86	33	12747
2015	4576	3017	1948	1237	929	425	361	297	114	49	12953
total	13480	11490	8539	4688	3793	1707	1608	1077	382	139	46903

The following table shows the distribution of assessments across the different suicide risk categories.

Table 9 GRiST data distribution across risk levels

Suicide Risk	No of Assessments
1	13480
2	11490
3	8539
4	4688
5	3793
6	1707
7	1608
8	1077
9	382
10	139
Total	46903

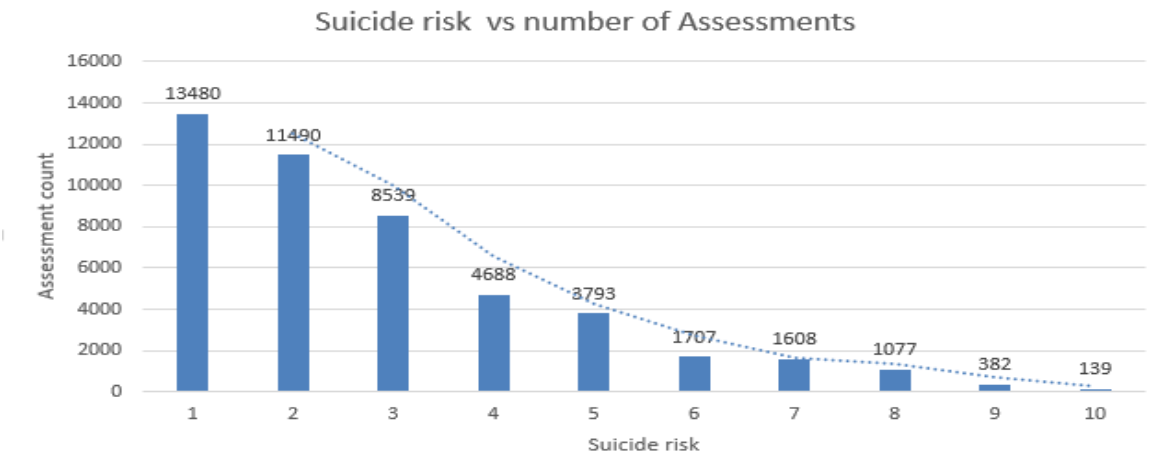

Figure 5 Suicide risk distributions in the GRiST dataset

The following table shows the number of assessments based on when they are conducted. The assessments conducted between 2011 and 2013 were used for training and rest were used for testing.

Table 10 Test and Training dataset from GRiST

Assessment year	No of Assessment	Used for
2011	4663	training
2012	6301	training
2013	10238	training
2014	12747	test
2015	12953	test

Textual comments input is optional in the GRiST system. The following figure shows a typical comments input field.

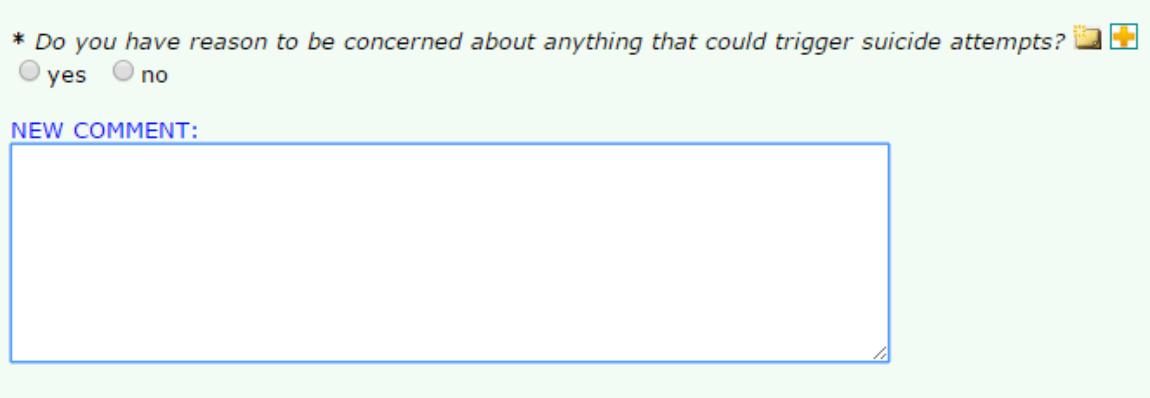

Figure 6 Example of a comment input field

The comments may be inputted in each of the GRiST nodes. Each node is a question that is asked by the clinicians or answered by a service user themselves. For our analysis, we have used comments left in the node to create the semantic vector of the nodes. All comments are concatenated to build the overall textual representation of the patients.

3.5 Improvement of the GRiST System

This research would try to improve GRiST's usability and interactivity by introducing background analysis and alerts. The following are the key areas where this research will add value to the system:

1. The comments in the GRiST nodes will be used to build a semantic vector representation of the GRiST nodes. This may allow us to use comments as another source of data in risk calculation. Concept extractions and text classification tools will be used against the GRiST dataset.

2. Explore the non-linear relationship between GRiST nodes and use them for system improvement. To find the relationships between nodes that may help us to detect high-risk category cases.

3. Explain the difference between clinicians and calculated risk. This may help us to guide the system users to improve the validity of their assessment. GRiST does not currently perform risk validation and this research has tried to add this feature to the GRiST system.

4. Make GRiST more interactive. Find out patterns in the data such as node relationships and provide notifications to clinicians based on the impact of those patterns on risk judgement.

5. Provide information to clinicians about risk management based on repeat assessment analysis.

As mentioned in the review of CDSS, service users prefer interactive CDSS. Before providing interactive suggestions to service users, we need to find out as many patterns as possible in the data. Then it would be possible to provide real-time notification or feedback to the clinicians based on the patterns found in the comments and inputted

data. The following diagram gives a very simplistic overview of the proposed system in comparison to the original system.

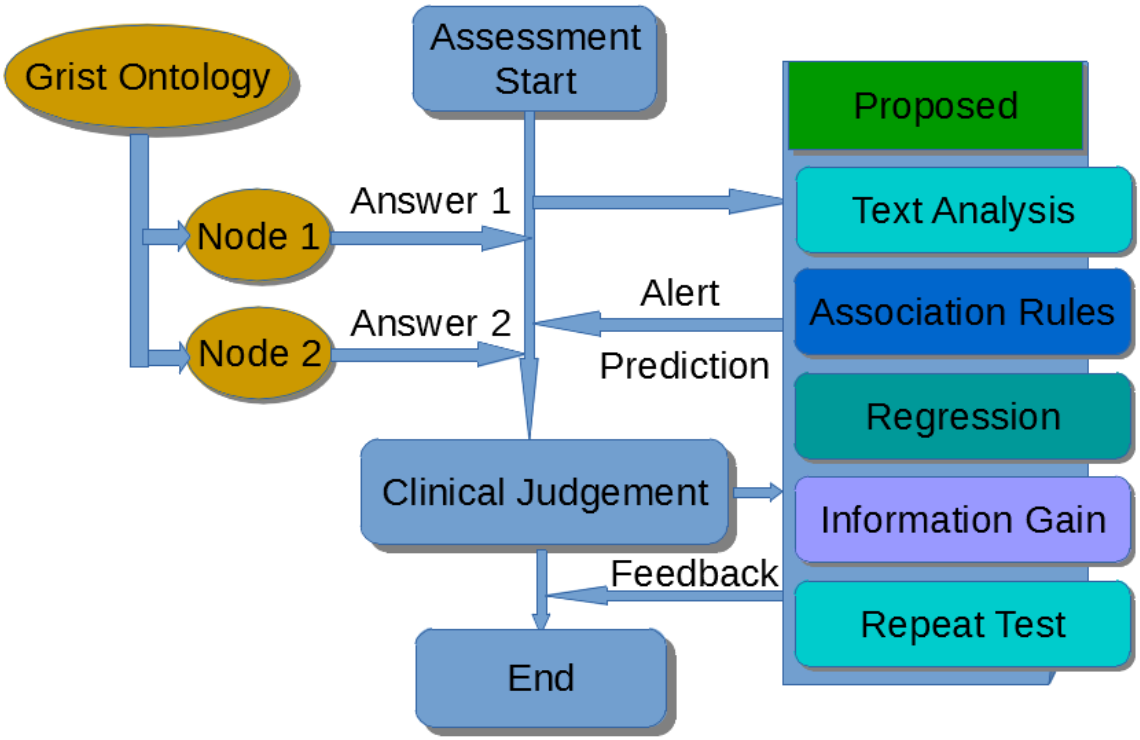

Figure 7 Aimed improvement of the GRiST system (in blue)

3.6 **Summary**

From the literature review, we have learned that the adoption of CDSS depends on accuracy, ease of use and decision supporting capability of the system (Kim et al., 2012). GRiST is a web-based clinical decision support system based on the Galatean Risk assessment model. This is an easy-to-use system and designed for experts, social workers and with service users in mind. It covers a total of five areas of mental health assessment, suicide, self-harm, self-neglect, harm to others and vulnerability.

Whilst GRiST was built on a clear research based foundation, the tool still lacks some desirable features. One of them is to carry out detailed background analysis of data and provide real-time feedback to the clinicians. It collects clinical comments, which are overlooked unless there is an emergency review. It is highly desirable to use artificial intelligence technology to make this system more interactive.

“Regarding the use of GRiST for predicting risks, its current objective is more about supporting clinical risk judgements and the associated advice rather than trying to output precise probabilities or some associated overall risk score. Scores and probabilities are extremely hard to produce with any accuracy or confidence, partly due to the current paucity of data but also because suicide, for example, is a very rare event”.
(Source: https://www.egrist.org/why-grist). This research also tried to overcome this limitation of GRiST.

We have tried to uncover as many ways as possible to enhance the GRiST system. Analysis of the free text comments available in GRiST, numerical data inputted by the clinician and GRiST ontology nodes relationships etc. all have been used to find patterns to improve GRiST.

Before using the comments, we need to extract concepts from them. The next chapter discusses concept extraction from text. The subsequent chapters focus on semantic processing of the textual data, suicide risk prediction, association rules mining and information gain. The methods proposed in those chapters can help to make the GRiST system an exceedingly interactive expert CDSS.

4 Concept Extraction

4.1 Introduction

Automatic extraction of concepts from the text is a prerequisite to many applications including information retrieval, text summarisation, question answering, classification, knowledge discovery and other natural language processing tasks (Aronson, 2006). Concepts from the clinical comments in GRiST were extracted for knowledge discovery and semantic analysis of the GRiST nodes.

Medical notes contain invaluable information about the current and previous medical history, current symptoms and severity of the condition as well as physicians clinical judgement (Cobb, Puri, Wang, Cise, & Edu, 2013). The GRiST data includes numerical and categorical input as well as free text comments. Processing of clinical comments may complement the collected numerical data. Many existing technologies and toolkits have been used and their performances have been critically reviewed.

Extracting domain relevant concepts is challenging. We need to extract phrases and then only keep those that are domain relevant. Generally, this is done by training a system with the human annotated dataset. This is time-consuming and costly. We propose a new method by which we extract domain relevant phrases automatically. The empirical results show that the proposed method could perform equally well as the systems that are trained on annotated data.

We have also extracted SNOMED-CT concepts extracted using cTAKES and compared its performance with other phrase extraction methods. A novel method has been described, which uses vector space model (word2vector) to automatically find a list of domain relevant keywords from the text. The proposed methods can save time and resources. The proposed Ensemble Concept Mining (ECM) method has been validated with both our own and the i2b2 dataset.

4.2 Phrase Extraction Methodology

The terms ‘phrase extraction’, ‘concept extraction’ and ‘symptom extraction’ are used synonymously in this report. We wanted to extract phrases that provide cues for medical symptoms, procedures or any other information that might help to assess patient’s medical conditions. Key phrase extraction is a technique that is used often for content summarisation, indexing, and information extraction. Where possible existing tools and methodology were used, and more focus was placed on GRiST ontology analysis. The following sections describe some of the common key phrase extraction methods. All of these methods have been applied to GRiST data.

4.2.1 N-gram Method

This is the most basic approach of phrase extraction. This method is chosen as a baseline method for phrase extraction. Here ‘n’ stands for the number of words in the phrase.

Sentence: David is depressed.
Unigrams: David, is, depressed
Bigrams: David is, is depressed

Understandably, not all n-gram phrases are a real or meaningful phrase, so a filtering algorithm is needed. Most of the other methods use n-grams at the beginning and then filter the generated phrases to find suitable phrases. Few basic filtrations can be done on n-gram phrases, for example, removing phrases that start with a preposition or only contain stop words (Pudota, et al. 2010).

4.2.2 Term Frequency

The score of a candidate term is related to the frequency of that candidate term occurring in the corpus (Justeson & Katz, 1995). “The great majority of technical terms

are noun phrases, largely limited to those including adjectives and nouns only. In running text, most topically important technical terms are repeated; those noun phrases that are repeated are very likely to be technical terms" (Justeson & Katz, 1995, p. 24).

4.2.3 TF-IDF

Equally frequent terms do not mean they are equally meaningful and another term IDF (Inverse Document frequency) is used to compensate for this (Church & Gale, 1999). TF-IDF stands for Term Frequency-Inverse Document Frequency, which is a statistical measure used to calculate how a word is important to a document in comparison to a corpus. The importance increases if a word appears more in the document but if the word generally appears more in the corpus then the weight is reduced.

TF (Term Frequency) represents the number of times the phrase appears in the document. DF (Document Frequency) represents the number of documents containing the phrase in the corpus. TF-IDF score is computed as

$$TFIDF = \frac{TF}{Nd}\ x\ log\frac{Nc}{DF} \quad (1)$$

Where Nd is the number of words in a document and Nc is the number of documents in the corpus (Jiang, Hu and H. Li, 2009).

4.2.4 Weirdness

The concept of Weirdness described by Ahmad, Gillam, & Tostevin (1999) as follows:

> *The differences in the distribution of certain lexical items, and their variants, in special and general language texts can be quantified in terms of the relative*

frequencies of a specialist text (corpus) and a general language text corpus. We call this ratio an index of weirdness of a specialist text. This weirdness is used by an accentuated, and perhaps an eccentric, choice of lexical items measured in terms of their frequency of occurrence. Most weird words in a text will tend to represent it more closely than those that are not as weird. If the ratio is unity, then the lexical item has the same frequency in both general and special language; if the ratio is greater than unity then the item is used more frequently in specialist text then is the case for general language and vice versa. (p. 4)

$$Weirdness = \frac{ws/ts}{wg/tg} \quad (2)$$

Where:

ws = frequency of word in specialist language corpus
wg = frequency of word in general language corpus
ts = total count of words in specialist language corpus
tg = total count of words in general language corpus

4.2.5 C-value Method

The C-value method combines linguistic and statistical information, with greater emphasis on the statistical part. The linguistic information may consist of part-of-speech tagging, types of term extracted. The statistical part takes into account the statistical features of the candidate string and combines it in a form of measure that is called C-value (Frantzi, Ananiadou, & Mima, 2000).

The C-value statistical measure ascribes a term hood to a candidate string, which is used to rank the output list of terms. The measure is built using statistical characteristics of the candidate string and they are as follows (Frantzi et al., 2000):

1. The total frequency of occurrence of the candidate string in the corpus.

2. The frequency of the candidate string as part of other longer candidate terms.
3. The number of these longer candidate terms.
4. The length of the candidate string (in number of words).

C-value is a domain-independent method for multi-word term extraction which aims to increase the extraction of nested terms (Frantzi et al., 2000). The measure of term hood, called C-value is calculated by the equation below:

$$C\text{-}value(a) = \begin{cases} \log_2|a| . f(a) & \text{if } a \text{ is not nested,} \\ \log_2|a|\left(f(a) - \frac{1}{P(Ta)} \sum_{b \in Ta} f(b)\right) & \text{otherwise.} \end{cases} \quad (3)$$

Where:

a = is the candidate term,
b = longer candidate terms
|**a**| = length of candidate term
f (a) = frequency of occurrence of **a** in the corpus,
f(**b**) = frequency of occurrence of **b** in the corpus,
Ta = is the set of extracted candidate terms that contain **a**,
P(Ta) = number of candidate terms in Ta.

The above formula is taken from the paper (Frantzi et al., 2000). It can be seen that C-value depends on the frequency of occurrence of a term but if the candidate term appears in the longer terms then this has a negative effect (Frantzi et al., 2000).

4.2.6 GlossEx

GlossEx is an algorithm designed to extract domain specific glossaries from large document corpus (Park, Byrd, & Boguraev, 2002). The following are the steps used to find the candidate terms.

Step1: Identify glossary terms. They considered either a noun phrase or a non-auxiliary verb.

Step 2: Filter terms bases on symbolic variants, compounding Variants, inflectional variants, misspelling variants, and abbreviations.

Step 3: Glossary item ranking and selection.

Candidate terms are then ranked based on the goodness of the term and the goodness of the term is based on how much the term is related to the given domain, the item's domain-specificity, and the degree of association of all words with the term (called term cohesion) (Park et al., 2002). The term confidence of a term T, C (T), is defined as below

$$C(T) = \alpha * TD(T) + \beta * TC(T) \quad (4)$$

Where TD is term domain-specificity, TC is term cohesion, and α and β (α+β=1) are constant values, which decide the relative contribution of TD and TC respectively (Park et al., 2002).

4.2.7 TermEx

The TermExtractor first extracts typical terminological structures, like compounds (enterprise model), adjective-noun (local network) and noun-preposition-noun (board of directors) and then uses filters to exclude non-terminological multi-word strings from the list of syntactically plausible candidates (Sclano & Velardi, 2007).

The algorithm used three main filters and two additional filters. A short description of them is given below, which is taken from the main paper (Sclano & Velardi, 2007).

Domain Pertinence: This value is high if a term is frequent in the domain of interest and much less frequent in the other domains used for contrast.

Domain Consensus: This value measures how evenly the term is distributed in the documents set.

Lexical Cohesion: This measures the degree of cohesion among the words that compose the term. The cohesion is high if the words composing the term more frequently appears in the documents.

The algorithm then uses some heuristic information to enhance the filters, such as structural relevance, misspelling, etc.

Structural Relevance: This fine tune the measurement by looking to see if the word appears in the text highlighted are also appears in title etc.

Miscellaneous: A set of heuristics are used to remove general modification of terms, misspelling, etc.

The final weight of a term is a weighted linear combination of the three main filters (Sclano & Velardi, 2007).

4.2.8 Sentic Algorithm

The first step of this algorithm is to break text into clauses by using dependency parsing. Then each verb and its associated noun phrase are considered in turn, and one or more concepts are extracted from these (Rajagopal, Cambria, Olsher, & Kwok, 2013). They have used different patterns like (adj+noun) to filter the concepts. According to them in Human Computer Interaction (HCI) and social data analysis, deep natural language understanding is not strictly required. A sense of the semantics in text and effects of that semantics is often enough to quickly perform tasks such as emotion recognition and polarity detection (Rajagopal et al., 2013). Further information about this method can be found in (Poria, Cambria, Winterstein, & Huang, 2014).

The Java implementation of this algorithm is fast and can work on a single sentence unlike other toolkits that require a complete document or corpus to extract phrases. This algorithm has been used to extract phrases from the GRiST data.

4.2.9 Other ATE Methods

There are other well-known Automatic Term Extraction (ATE) algorithms for example, Kea, Topia, Termine and AlchamyAPI, which were reviewed but not used. Firstly, some of them extract important key phrases for text summarisation purposes only. For this research, all the concept phrases are considered important. Secondly, some of them are only available as web service and due to the confidential nature of our data, it was not possible to use them. Uses of dummy data did not show any significant improvement over the algorithm that has been tried in this research. A review of recent automatic term extraction can be found in (Siddiqi & Sharan, 2015), (Z. Zhang, Iria, Brewster, & Ciravegna, 2008) and (Z. Zhang, Gao, & Ciravegna, 2016).

For this research, I have used all the methods that are generic and not specially designed for text summarisation. Their performances have been evaluated with a sample manually extracted concept list. The evaluation and analysis of the results are provided later in this chapter.

4.3 UMLS and SNOMED-CT Concept Extraction

The Unified Medical Language System (UMLS) consists of a set of files and software utilities that merge together many health and biomedical vocabularies and standards to enable interoperability between computer systems (Source: https://www.nlm.nih.gov/). SNOMED-CT (Systematized Nomenclature of Medicine - Clinical Terms) is a standardized, multilingual vocabulary available from SNOMED International (http://www.snomed.org/snomed-ct) and it is included in UMLS. In this report, the term

UMLS or SNOMED-CT are sometimes referred to as “snomed” to mean a terminology database.

According to UMLS reference manual (National Library of Medicine, 2009) and website (https://www.nlm.nih.gov/research/umls/quickstart.html), The UMLS has three tools, which are called the Knowledge Sources:

1. Metathesaurus: Terms and codes from many vocabularies, including CPT, ICD-10-CM, LOINC, MeSH, RxNorm, and SNOMED-CT.

2. Semantic Network: Broad categories (semantic types) and their relationships (semantic relations).

3. SPECIALIST Lexicon and Lexical Tools: Natural language processing tools.

4.3.1 Metamap

Metamap is a widely available program that can discover Unified Medical Language System (UMLS) Metathesaurus concepts referred to in the biomedical text (Aronson & Lang, 2010). This is particularly useful as the discovered concepts would be more relevant to mental health or generic medical concepts. This tool also provides UMLS node names and that could allow us to compare it with the GRiST nodes.

Metamap uses knowledge base (ontology), natural-language processing (NLP), word sense disambiguation (WSD) and computational-linguistic techniques. More information can be found in (https://metamap.nlm.nih.gov/). A survey of direct uses of SNOMED-CT can be found in (Elhanan, Perl, & Geller, 2011). To use Metamap and UMLS a registration is required.

For this research, Metamap was installed locally and the extracted sentences from each assessment were fed into it via a command line program. The output was then saved in the database table.

4.3.2 Apache CTAKES

Apache CTAKES is another well-known tool for mapping text to SNOMED-CT ontology. Popular open source tools like General Architecture for Text Engineering (GATE) have plug-ins for this task. An automated system for conversion of clinical notes into SNOMED-CT clinical terminology is described by Patrick, Wang, & Budd (2007) and a method for encoding clinical datasets with SNOMED-CT is described by Lee, Lau, & Quan (2010). Both of these researches used string matching and heuristic rules.

For this research, I have downloaded and installed the apache cTAKES tool locally (available from http://ctakes.apache.org/). Registered an account with UMLS (https://www.nlm.nih.gov/research/umls/) and downloaded UMLS terminology dataset. Then I have coded a Java program to extract UMLS coded XML data from a given sentence. Then phrases were extracted based on some filtering on the concept type. The output was the phrases that mapped to a SNOMED-CT concept. The full process is described in Chapter 5.

cTAKES was used as it is new and provides more detailed information in XML format. The SNONED-CT data was saved in a local database and we could traverse through the ontology tree to explore the data and link them to GRiST nodes.

4.4 Word Embedding

Word embedding is a technique where a word is represented by a vector of real numbers. In general, it is also referred to as vector space model (VSM). A hypothetical example could be:

‘depression’=[0.4, 0.3,0.6,0.8,0.7]

Vector space models (VSM) of word co-occurrence have proved to be a useful framework for representing lexical meaning in a diverse range of natural language processing (NLP) tasks (Padó & Lapata, 2007). Most of the VSM algorithms are based on distributional semantics, which simply means words that appear in similar context

tend to have a similar meaning (Turney, 2013). Vector-based semantic space models use word co-occurrence counts from large corpora to represent lexical meaning (Padó & Lapata, 2007). The context of the target word is defined by a small number of words surrounding the target word. It does not even consider the parts of speech of the words. For this research the word2vec tool developed by Mikolov, Yih, & Zweig (2013) is used. There are lot of recent activity around this tool and huge trained datasets are readily available. A detailed explanation of this tool and data is given in the following section.

4.4.1 Word2Vec

Many methods of creating word representations were explored by the NLP community (Levy & Goldberg, 2014). Continuous word representations, trained on large unlabelled corpora are useful for many natural language processing tasks (Mikolov, Yih, et al., 2013). The word2vec model and its application by Mikolov et al. (2013) have attracted a great deal of attention in recent years. The vector representations of words learned by word2vec models have been shown to convey semantic meanings and are useful in different NLP tasks. Distributed representations of words in a vector space can facilitate grouping of similar words and help learning algorithms to achieve better performance in natural language processing tasks (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013).

In word2vec, a distributed representation of a word is used. Take a vector with several hundred dimensions (say 300). Each word is represented by a distribution of weights across those elements. So instead of a one-to-one mapping between an element in the vector and a word, the representation of a word is spread across all of the elements in the vector, and each element in the vector contributes to the definition of many words (Mikolov, Yih, et al., 2013).

Table 11: Word vector example

Word	Attribute 1	attribute 2	attribute 3	attribute 4	attribute 5
King	0.3	.25	0.36	0.36	0.40
Queen	0.3	.36	0.36	0.20	0.30

The “google news model” that is used in this research has 300 dimensions in total (attribute1,… attribute300) - the above table is only showing 5 dimensions as an example. We have also used vector trained on PubMed text, which represents the medical domain. A defining feature of language models which uses the neural network is their representation of words as high dimensional real-valued vectors (Mikolov, Yih, et al., 2013).

Word2vec is an algorithm that learns the representations of the word vectors, which captures the semantic relationship in an unsupervised way (Mikolov, Corrado, Chen, & Dean, 2013). According to them Word2Vec is a shallow learning algorithm that has 2 variations through which it learns the word vector representation using neural networks:

Skip Gram Model: In this method given the word, the system tries to predict context words. Context words do not need to be immediate words, some words can be skipped, setting a window size determines how much to look forward and backwards from the current word (Mikolov, Corrado, et al., 2013).

Continuous Bag of Words (CBOW): Given the context word, the target word is predicted. The big window size improves semantic scores but reduces the syntactic score. For CBOW the size of training data should be fairly large enough as a single word is predicted from many contextual words (Mikolov, Corrado, et al., 2013).

Figure 8 Word2vector training algorithm types, reproduced from (Mikolov, et al. 2013)

The above image is reproduced from (Mikolov, Corrado, et al., 2013). Further details of word2vec can be found in (Mikolov, Corrado, et al., 2013), (Mikolov, Yih, et al., 2013)

and (Mikolov, Sutskever, et al., 2013). In this report, term ‘word vector’ is used many times and it generally means the vector representation of a word using word2vec.

4.4.2 Phrase Vector

A phrase may contain two or more words. To make a phrase vector from word vectors Kiela & Clark (2013) used pointwise multiplication and Lopyrev (2014) used neural network re-training. The experimental results indicate that the simple average of word vectors may perform well enough for generic use (Lopyrev, 2014). In this research, the simple average method was used to calculate phrase vectors from words vector.

4.4.3 Cosine Similarity

The output of the Word2vec is a vocabulary where each word is represented by a vector. The vectors carry semantics meaning and similar words appear closer in the vector space. A variety of distance measures can be used to compute the similarity between two target words, the cosine similarity is the most popular (Padó & Lapata, 2007).

The cosine similarity measure is the cosine of the angle between two vectors. In this measure, 90 degrees represent no similarity cos (90) =0 and the total similarity is 0-degree angle cos (0) =1. Given two vectors X and Y we can calculate cosine similarity by using the following formula (Padó & Lapata, 2007).

$$CosineSimilarity(cos\theta) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} . \sqrt{\sum_{i=1}^{n} y_i^2}} \qquad (5)$$

The following table shows some words and their distance from the word ‘drug’.

Table 12 Similar word and cosine distance

Word	Distance
Drug	1.000
Drugs	0.849
narcotics	0.643
cocaine	0.609
Heroin	0.608
medication	0.542
Pills	0.521
Dope	0.511
marijuana	0.503
Pill	0.502

For this research I have used word2vec tool developed by Mikolov, Corrado, et al. (2013) and available from (https://github.com/dav/word2vec).

The word vectors were trained on part of Google News dataset (about 100 billion words). The model contains 300-dimensional vectors for 3 million words and phrases (Mikolov, Sutskever, et al., 2013). A wrapper around this library was created so it can be easily used as a web service from the Java and PHP based tools that have been created to analyse the GRiST data. Later I have also used vector trained with PubMed data available from (http://evexdb.org/pmresources/vec-space-models/).

4.5 Evaluation of Phrase Extraction Methods

Initially, many other tools have been used for test purposes but in the end, they were not utilised. Stanford NER (Name Entity Recogniser) was used to extract noun phrases (Finkel, Grenager, & Manning, 2005). I have assumed that valuable clinical information

could be in ‘verb’ as well as in ‘noun’ phrases, which is why NER was not used in isolation. The datasets, experiments and evaluations that have been used are explained in the following sections.

4.5.1 Experimental Procedure

I have run all of the phrase extraction algorithms as discussed above on the GRiST dataset. The evaluation was done on a small subset. The following are the generic steps of the operation:

Step 1: All the comments from each GRiST node have been saved into a database table. Comments are then split into sentences and stored in a separate table keeping the record of assessment_id and GRiST node name where the comment was found. The node name is sometimes referred to as ‘nodecode’ in this report, as this is a unique identifier of a node.

Step 2: In the next step sentences were parsed with Enju parser (available from http://www.nactem.ac.uk/enju/) and Stanford parser (Manning, Bauer, Finkel, & Bethard, 2014). Outputs of these parsers provide the base form of the word. The Stanford parser was chosen as it also provides the dependency relationships.

Step 3: A web service was created using Java for the phrase extraction from sentences. These web services work as a wrapper around the Java jar file that we have downloaded for each of the algorithms. This allowed us to use many algorithms to create phrase extractors and consume them as a web service. Java was good for parsing and PHP was good for rapid prototyping - this is why both of them were used in this manner.

Step 4: PHP was used as a scripting language to consume the web services to extract phrases and put them in a separate database table. Again, the record of assessment_id, sentenceid, nodecode etc. was kept for future reference.

4.5.2 Statistical Evaluation

The accuracy of key phrase extraction methods is generally evaluated by precision and recall measures, for which human annotated phrases are treated as positive examples (Xin Jiang et al., 2009).. We have used precision, recall and F-score. Before showing the data, here is an introductory explanation of each of these measures.

Precision, Recall and F-measure are standard measures in Natural language processing (Manning, Raghavan, & Schütze, 2009). The following figure shows a typical phrase extraction scenario.

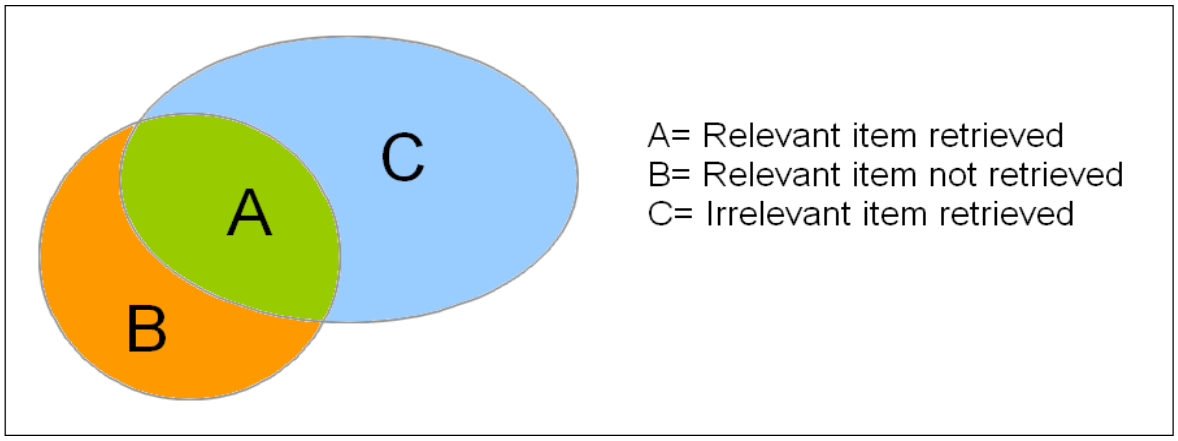

Figure 9 Precision and Recall

Precision: PRECISION is the ratio of the number of correct records retrieved to the total number of incorrect and correct records retrieved. It is usually expressed as a percentage.

$$Precision(P) = \frac{A}{A + C} \quad (6)$$

Recall: RECALL is the ratio of the number of correct records retrieved to the total number of correct records in the dataset. It is usually expressed as a percentage.

$$Recall(P) = \frac{A}{A + B} \quad (7)$$

F-measure: F-measure is a single statistical measure that trades off precision versus recall, it is the weighted harmonic mean of precision and recall (Manning et al., 2009):

$$F_{measure} = \frac{(\beta^2 + 1)pr}{\beta^2 p + r} \quad (8)$$

The default balanced F-measure equally weights precision and recall, which means making β = 1. It is commonly written as F1, which is short for Fβ=1. When using β = 1, the formula on the right hand side simplifies to (Manning et al., 2009):

$$F_1 = 2 * \frac{pr}{p + r} \quad (9)$$

Another concept Error Rate, is the percentage of examples that are assigned to the wrong category (Celikyilmaz, Hakkani-Tur and Feng, 2010).

$$Errorrate = 100 - accuracy \quad (10)$$

Often accuracy is not a suitable measure for information retrieval problems as most of the cases, the data can be extremely skewed: normally over 99.9% of the documents are in the non-relevant category (Manning et al., 2009). For example, maximum patients are in the low suicide risk category so predicting all patients of having low risk would give a highly accurate result, but it would miss all the high-risk patients.

4.5.3 Results and Analysis

It is very important to validate the extracted concepts. An algorithm may perform well in one set of data but not so well in another dataset, especially if the domain context is different. Creating a gold standard data is a manual task and requires multiple people. There are more than a million phrases extracted so manually validating them was not feasible. I have taken approximately 20 thousand of them from a set of assessments

ranging between 2010 to 2011 and manually marked them to check if they were relevant to suicide risk or not.

The phrases were filtered and only bigrams (2 word phrases) were used. There were too many unigrams and a reasonable assumption was that we could create a domain specific unigram list by frequency analysis (described later in section 4.6). It was observed that bigrams are more practical and meaningful to use. A few other research papers mention that bigram phrases are more suitable including (Xuerui Wang, McCallum, & Wei, 2007). The single word and multi-word phrases were extracted and saved in the database but validation was done only on bigrams because a single word may apply equally well in many other contexts, whereas multi-word phrases were expected to be more semantically concise in their usage (McCart et al., 2012).

This was a preliminary exercise to explore the accuracy of the existing methods. Based on the knowledge and understanding that I have acquired after looking at the GRiST ontology, the risk assessments data and reading more on this subject matter, I filtered the extracted phrases based on its relevance to suicide risk. The performance on this dataset should give us an indication of the suitability of the algorithms. The following table shows the test results with precision, recall and F1-score. Please note that the phrase count was done on distinctive phrases only.

Table 13 Precision and recall of phrase extraction algorithms

Algorithm	Count	Right_A	Missed_B	Wrong_C	Precision	Recall	F1-score
n-gram	938	370	457	568	0.39	0.45	0.42
b-gram	462	196	631	266	0.42	0.24	0.31
metamap	40	28	799	12	0.70	0.03	0.06
Sentic	1500	488	339	1012	0.33	0.59	0.42
jatetfalgo	1000	320	507	680	0.32	0.39	0.35
jatecvalue	1315	376	451	939	0.29	0.45	0.35
jateglossex	968	336	491	632	0.35	0.41	0.38
jatetermfreq	908	300	527	608	0.33	0.36	0.34
jatetfidf	812	288	539	524	0.35	0.35	0.35
jateweirdness	818	286	541	532	0.35	0.35	0.35

We can see from the above table that the metamap tool has the highest precision. But unfortunately, it has the lowest recall. Metamap maps phrases to the UMLS metathesaurus and it only looks at those terms, which are available in UMLS. Hence, the recall is very low, as it does not extract many other possible terms.

All other algorithms have very low precision and recall as well. They are usually in the range of 30% to 40%. This could be because these algorithms are trained with different kinds of data. An algorithm usually performs better on the domain that it has been trained. It is highly desirable to have a generic algorithm that can be used in multiple domains without retraining.

The analysis shows that none of the tested algorithms can perform as desired with our data. Some of these algorithms are very complex, but when we compare their results with simple n-gram extracted phrases we see that the n-gram also performs equally well. From this, we can assume that the use of sophisticated algorithms may increase the complexity and make the extraction process slow, but the overall performance does not improve significantly. This finding matches with the findings of (Rajagopal et al., 2013).

The extracted phrases were not always meaningful to the suicide context. After obtaining poor results from the existing algorithms, a decision was made to extract n-grams and then filter them to extract relevant phrases. The proposed new phrase extraction and filtering method is described below.

4.6 The Ensemble Concept Mining (ECM) Method

Existing automatic phrase extraction methods produced very low precision results as shown in the previous section. Training a model with human annotated data is very resource extensive. To improve the accuracy of extracting domain relevant phrases, we have considered a new phrase extraction and filtering method. We have hypothesised that we may extract phrases by using simple n-gram or other methods and then filter them for domain relevancy by comparing their semantic distance from a domain relevant

word list. A new Ensemble Concept Mining (ECM) method has been proposed for this purpose. The following sections describe the proposed ECM method in detail.

4.6.1 Description of the ECM Method

A concept phrase or key phrase needs to have two features: phraseness and informativeness (Tomokiyo & Hurst, 2003). Phraseness is the degree to which a word sequence is considered to be a phrase. Informativeness refers to how well a phrase relates to a specific domain. The proposed method covers both of these ideas. It is an ensemble method, which means it takes ideas from other methods and adapts them for the GRiST data. It uses word embedding for the domain relevancy filtering purpose.

The method has two parts. The first part is the extraction of key phrases by using rules of phraseness and the second part is the semantic based domain relevancy filtering for the domain relevancy. The first part can be replaced by any other generic key phrase or term extractor if need be. The fundamental idea is to use a generic phrase extraction method and then apply semantic vector based domain relevancy filtering.

Figure 10 Ensemble Concept Mining (ECM) Method

Part one: Phrase Extraction

In this step, the phrases are extracted from the biomedical data. The extracted phrases grammatically must resemble a phrase.

1. Sentences are extracted from the text as it is assumed that no key phrase parts are located simultaneously in two sentences (Pudota, et al. 2010). Special symbols such as '.', '@', '&', '/', '?', '!' were replaced with sentence delimiters. N-gram phrases are then extracted from sentences.

2. It has been noted that a phrase usually does not start or end with a common preposition (El-Beltagy & Rafea, 2009). A list of English prepositions was created and if the phrases match with them or start with them then it was discarded.

3. Stop words filtering. A list of English stop words was created. If the phrase is a stop word or if it contains only stop words, then it may not be a phrase. A list of stop words can be found in Stanford CoreNLP toolkit (Manning et al., 2014).

4. The word that expresses meaning (concept words) tend to be longer (Joos & Zipf, 1936). A domain specific exceptional concept word list could be created. Words not in the exception list but less than 5 characters long are dropped.

5. A phrase usually does not start with a number; in fact, in this case all numbers containing phrases are dropped. This option can be adjusted as per requirement.

6. Words in the concept phrase appears in other phrases (Parameswaran et al., 2010). For example, a single concept word may appear in bigrams. This idea comes from the C-value method (Frantzi et al., 2000).

7. Phrases usually carry a noun word in them (Subhashini & Kumar, 2010). If at least one word is not a noun, then it can be dropped. This is an optional step and we have considered a verb with an adverb could also be a concept. We need to parse the sentence to get POS tags. One can consider many other combinations of word types as given in (Siddiqi & Sharan, 2015).

8. How frequently the phrase appears in the document is used as a feature. A concept phrase should not be rare in the document (Bleik et al., 2010). A cut

off point can be set. It is not applicable in our case as this is mainly applicable in text summarisation.

9. Phrase spreads in multiple documents. In this case, a key phrase should appear in more than one patient's assessment. Originally it was presence of phrase in multiple documents in a corpus (Pudota, et al. 2010). We can set a cutoff point for this.

Part two: Semantic Domain Relevancy Filtering

This step semantically filters the phrases for domain relevancy. Even if a phrase is grammatically correct but not semantically related to the current context then it is removed.

10. Extract the domain relevant word list by comparing the frequency of the words with a reference corpus such as Wikipedia text data.

11. A concept phrase should be semantically related with other phrases of the context. We can use word vectors and cosine similarity measures to semantically filter phrases and retain the domain relevant phrases.

The above steps 10 and 11 of domain relevancy filtering technique is described in detail in the next section. Pudota, et al. (2010) proposed a technique to combine different phraseness measures mathematically. A phraseness score is calculated as a linear combination of all the features. The phraseness of a phrase P with a non-empty feature set {f1,f2,...,fn}, with non-negative weights {w1,w2,..,wn} is:

$$Phraseness(P) = \frac{\sum_{i=1}^{n} w_i . f_i}{\sum_{i=1}^{n} w_i} \quad (11)$$

The proposed ECM method adopts ideas from other algorithms and incorporates measures that are more suitable for medical notes and GRiST comments. For example, some algorithms argue about the appearance of a word in the beginning or end of a document, which may affect its ranking. But for clinical comments, it is not possible to apply that rule, as the data may be stored in separate database table fields. In the case of GRiST, comments were stored in a specific ontology node.

4.6.2 Automatic Domain Relevancy Filtering

Extracting phrases relevant to a specific domain automatically can be challenging. We assumed that we might create a list of the most frequent words and then use semantic similarity measures to detect conceptually similar phrases. In the first stage, we can use traditional frequency based analysis to retrieve words and in the second stage, word vector based cosine similarity may determine the relevancy of the phrase. The proposed two-stage method is described below.

Automatic Relevant Word Extraction:

Step1: Create a list of words that are more frequent in the domain than in a reference corpus.
Step2: Calculate word vectors for each of the words in the list.
Step3: Count the number of similar words in the list for each of the words.
Step4: Sort the words based on the similar word count scores and extract a certain number of them.

Filtering for Domain Relevancy:

Step1: Take a key phrase and calculate its word vector.
Step2: Calculate cosine similarity between the key phrases and the relevant word list.
Step3: If the similarity is greater than a certain value, then mark the key phrase as relevant.

The above method was applied to find a list of keywords automatically. It shows promising results when tested with I2B2 data, which is described later in this chapter.

4.7 Validation by using the GRiST dataset

Each of the components of ECM is validated by relevant previous research as referenced to in the previous description of the method. Clinical comments from each of the GRiST nodes was extracted and concatenated as if each assessment was a document. Then the comments were split into sentences as per the delimiter defined in ECM. These sentences are then parsed by the Stanford parser (Manning et al., 2014) and saved in the database.

With a PHP script, these sentences are then split to create unigram, bigram and trigram. Bigrams provide more specific meaning and they have been used extensively in concept extraction and text classification research (Bekkerman & Allan, 2003), (Xuerui Wang et al., 2007), (Hasan & Ng, 2010) and (Justeson & Katz, 1995). For validation purposes, I have only used bigrams. Due to the confidential nature of the data and lack of resources manual annotation by multiple people was not possible. Initially, validation was done by using self-annotated data and then with the i2b2 and semeval2010 dataset.

For step 10 of the ECM, a semantic filter was created with the metamap extracted key phrases. It has been found that metamap extracted bigrams are more meaningful or context related. At first the semantic vector was calculated for each metamap bigrams and then cosine similarity of the selected phrase with any one of the metamap bigrams was calculated. If similarity was less than 0.80 (it is a parameter and chosen by trial and error) then the phrase was discarded. Semantic vector based similarity is described in chapter 4 in detail.

For a total of 4018 assessments, there were 150,083 bigram phrases. After applying the ECM, 6056 phrases were found to be relevant. The following table shows some bigrams without ECM applied, with ECM applied but not including semantic filter (step 11) and with ECM applied including semantic filtering.

Table 14 Results of Phraseness Algorithm

No ECM Phraseness filtering	**ECM Phraseness**	**ECM with Semantic filter (yes means relevant)**
this have	feel suicidal	feel suicidal=yes
have occur	poor memory	poor memory=yes
occur but	further exploration	further exploration=no
but currently	serious suicide	serious suicide=yes
currently this	suicide attempt	suicide attempt=yes
this be	fractured spine	fractured spine=yes
be approximately	bubble bath	bubble bath=no
approximately twice	slashing wrist	slashing wrist=yes
twice a	unprovoked assault	unprovoked assault=yes
a week	threatening throughout	threatening throughout=yes
suicidal because	verbally hostile	verbally hostile=yes
because of	florid psychotic	florid psychotic=yes
of have	physically intimidate	physically intimidate=no
have ms	overnight leaf	overnight leaf=no
ms and	lewd conversation	lewd conversation=yes
and feel	continued thought	continued thought=yes
feel everyone	living skill	living skill=yes
everyone would	tissue damage	tissue damage=yes
would be	drinking excessively	drinking excessively=yes
be good	drink bleach	drink bleach=yes
good off	suicidal thought	suicidal thought=yes
off without	learning disability	learning disability=yes
without her	racial comment	racial comment=no
information at	damage property	damage property=yes
at time	easily influence	easily influence=yes
time of	mild learning	mild learning=yes
of need	self destructive	self destructive=yes
need to	permanent foster	permanent foster=no
to be	emotional abuse	emotional abuse=yes
be look	alcohol relate	alcohol relate=yes

From the above data, we can see that ECM (without step 11 semantic filtering) discards lots of simple phrases like ‘of have’, ‘feel everyone’ etc. However, it retains phrases like ‘overnight leaf’, ‘inpatient setting’ etc. Clearly using just syntactic filters is not enough as the phrases in column two in the above table demonstrates. When a semantic filter (step 11) is applied, it removes ‘overnight leaf’. We can see bigrams such as ‘physically intimidate’ is also filtered out. The results can be improved by changing parameters (i.e. minimum matching level) and adding more quality phrases to the predefined domain phrase list. Further validation was carried out with i2b2 (2010) dataset, which is described in the following section.

The proposed ECM uses some rules that were developed and analysed by previous researchers. References to them are provided in the description where appropriate. My contribution was to adopt key parts of the other methods and modify them for GRiST data (steps 5 and 9). Steps 6 and 7 are simplified and made optional. I have added domain relevancy filtering (steps 10 and 11). The resulting final method is more readily applicable in a CDSS system like GRiST. The method is further validated by using the i2b2 dataset, which is described below.

4.8 Validation by using the I2B2 dataset

To validate the ECM method the i2b2 2010 challenge dataset was used. The 2010 i2b2/VA Workshop on Natural Language Processing Challenges for Clinical Records presented three tasks: a concept extraction task, which focused on the extraction of medical concepts from patient reports; an assertion classification task, which focused on assigning assertion types for medical problem concepts and concept relationships classification tasks (Uzuner, South, Shen, & DuVall, 2011);

The i2b2 datasets are available to the research community at large from November 2011 from https://i2b2.org/NLP/DataSets. A total of 394 training reports, 477 test reports were de-identified and released. From the annotated text, we have removed phrases that contain numeric values so that we can calculate the word vector for each word. Wikipedia plain text was used as a generic reference text corpus. Wikipedia was chosen

as a generic corpus as it covers a wide range of topics and domains. It is also a very large and freely available dataset. The Wikipedia plain text corpus was taken from the URL (http://www.evanjones.ca/software/wikipedia2text.html).

Word vectors were induced from PubMed and PMC texts and their combination using the word2vec tool. The word vectors are provided in the word2vec binary format (PubMed-and-PMC-w2v.bin). The word vectors are available for download from the URL (http://evexdb.org/pmresources/vec-space-models/) (Pyysalo, Ginter, Moen, Salakoski, & Ananiadou, 2012).

4.8.1 Step1: Relevant Keywords Extraction

Let us consider that a domain can be represented by a list of words. A list of words was extracted from the I2B2 data by using the method described earlier in the “domain relevancy filter” section. Firstly, from the plain text Wikipedia corpus, word density (word occurrence count/total word count) was calculated. Then word density was calculated in the i2b2 training corpus (domain data). The words that are more likely to appear in the domain data are selected as domain specific words. In our experiments, we have chosen the words that are 10 times more likely to appear in the domain corpus and it must be present at least 5 times in each corpus to filter out very rare words. This was a choice taken based on the observation of the data. The choice may vary with another dataset.

In the next stage, we have calculated the word vector of each of these words. The cosine similarity was used to calculate the distance between words. The words that have a large number of matching words based on a minimum threshold are more likely to be domain specific words. We assumed that domain specific words would have many similar words present within the same domain corpus. We have achieved up to 0.66 f-score for finding the single keywords that were present in the annotation key phrases.

The following table shows some detailed results.

Table 15 Single word domain relevancy filtering results

Min match	Min Similarity	F-score
4	0.4	0.649
5	0.4	0.649
6	0.4	0.649
7	0.4	0.649
8	0.4	0.65
9	0.4	0.651
4	0.2	0.646
4	0.3	0.646
4	0.4	0.649
4	0.5	0.651
4	0.6	0.657
4	0.7	0.658

4.8.2 Step2: Validation and Comparison

To test the effectiveness of our method, i2b2 test and training data were used. To remain consistent with previous experiments we have again used only two words phrases. The sentic and n-gram methods that previously performed better with the GRiST dataset as well as RAKE and OpenNLP methods have all been used for comparison purposes.

1. OpenNLP: The Apache OpenNLP library is a machine learning based toolkit for the processing of natural language text and is available from (https://opennlp.apache.org). It implements a maximum entropy based phrase/name entity extractor, which can be trained with annotated text. We have trained OpenNLP with the i2b2 training dataset and used the trained model to extract phrases from the test dataset.

2. N-gram: In this case we simply extract phrases as n-grams and filter them by using the ECM method.

3. Chunk: In text chunking a text is divided into syntactically correlated parts of words, like noun groups and verb groups. We have used OpenNLP chunker to extract chunks and then filtered them by using the ECM semantic filtering method. OpenNLP chunker was trained on the conll2000 shared task data and a pre-trained English language model is available online from the URL (http://opennlp.sourceforge.net/models-1.5/).

4. Rake: Rapid Automatic Keyword Extraction (RAKE) is an unsupervised, domain-independent, and language-independent method for extracting keywords from individual documents (Rose et al., 2010). RAKE uses stop words and phrase delimiters to partition the document text into candidate keywords. Candidate keywords are then scored based upon co-occurrence, phrase length and frequency (Rose et al., 2010). A Python implementation available from the URL (https://github.com/csurfer/rake-nltk) was used.

5. Sentic: This algorithm breaks text into clauses by using dependency parsing. Then each verb and its associated noun phrases are considered in turn, and one or more concepts are extracted from these based on linguistic patterns (Rajagopal et al., 2013). A Java implementation of this algorithm was collected from the original authors.

The following two tables show some detailed results.

Table 16 i2b2 accuracy without ECM

system	Precision	Recall	F-score
ngram	0.15	0.99	0.26
chunk	0.39	0.73	0.51
opennlp	0.84	0.42	0.56
rake	0.50	0.57	0.56
sentic	0.16	0.55	0.24

Table 17 i2b2 accuracy with ECM

system	Precision	Recall	F-score
ngram	0.34	0.79	0.48
chunk	0.74	0.57	0.64
opennlp	0.90	0.33	0.48
rake	0.59	0.48	0.53
sentic	0.27	0.42	0.33

From the above data, we can observe that filtering with ECM improves the precision and overall f-score. The results of the RAKE and OpenNLP model did not improve as these methods already include some statistical relevancy filtering and adding further semantic filtering reduced recall. Chunking and then filtering with ECM produced the highest f-score (0.64).

The key difference between other methods such as Rake and ECM is that ECM uses semantic filter after applying rules and Rake and many others use statistical or co-occurrence measures. On the other hand, OpenNLP and many others use human annotated datasets for training a model. Sentic uses dependency parsing and common linguistic patterns to extract concepts.

From this, we can see that using word vector based semantic filtering after phrase extraction can improve accuracy. We may not need to create domain specific annotation. We can use a generic phrase extraction algorithm and then filter the output with wordvector based sematic filtering. This approach outperforms the OpenNLP (trained with domain phrases) method and the RAKE method (no need for training). This can save both time and resources required for annotation.

4.9 Semantic Phrase Ranking

Keywords highlight the main topic described in a document and its extraction is an important task in text analysis and information retrieval tasks. Algorithms such as RAKE (Rose et al., 2010) and others use statistical measures to rank candidate phrases. The common phrase ranking methods are term frequency, TF-IDF, term location, term length and co-occurrence with other terms etc. (Siddiqi & Sharan, 2015). We can modify part two of our ECM method and use it for semantic phrase ranking. In this case, we do not create a keyword list from domain corpus rather, we use semantic relatedness among candidate phrases to rank them.

Steps of Semantic Phrase Ranking:

Step1: Gather a list of Candidate phrases and calculate the semantic vectors for each of them.
Step2: Measure the sum of the cosine distances of a phrase from all the other phrases in the list.
Step4: Sort the phrases based on the score calculated in step 2 above and finally extract a certain number of them.

To validate our approach, we have used the semeval2010 dataset. RAKE is a well-known algorithm for extracting keywords (technically phrases) from a document by using word frequency and co-occurrence statistics, which is proposed by Rose et al.,(2010). Instead and in addition to the RAKE scoring method, we have applied our proposed semantic phrase ranking method on the semeval2010 key phrase extraction dataset.

The input parameters for RAKE comprise of a list of stop words (or stoplist), a set of phrase delimiters, and a set of word delimiters. RAKE uses stop words and phrase delimiters to partition the document text into candidate keywords. After every candidate keyword is identified, a score is calculated for each candidate keyword. They have used different measures such as (1) word frequency (freq(w)), (2) word degree (deg(w)), and (3) ratio of degree to frequency (deg(w)/freq(w)). In summary, deg(w) favours words that occur often and in longer candidate keywords. Words that occur frequently are favoured

by freq(w). Words that predominantly occur in longer candidate keywords are favoured by deg(w)/freq(w). The score for each candidate keyword is calculated from the sum of its member (Rose et al., 2010).

A Java implementation of the RAKE method as described by (Rose et al., 2010) is taken from (https://github.com/Linguistic/rake). For evaluation purposes, the SemEval-2010 Keyphrase extraction track data (a total of 144 documents) is taken from: (https://code.google.com/archive/p/maui-indexer/downloads#makechanges).

We have implemented a Java program in which RAKE keyword scoring was replaced by our own semantic similarity scoring method. A semantic similarity score is the sum of the cosine distance of a keyword from all other keywords in the candidate keywords list. We have also used a combination of semantic similarity and the RAKE method to score a keyword. The top 10 keywords are extracted and matched with the human annotated keywords in the semeval2010 dataset. The correctly extracted bigrams and trigrams are shown below.

Table 18 Semantic score and RAKE score extracted correct keywords

No of Documents	Human Annotated keywords	RAKE extracted	Semantic score extracted	Semantic +RAKE score
144	1684	176	246	272

From the above table we can see that if we use semantic scoring instead of the co-occurrence based scoring of RAKE, we can extract more keywords that are correct. If we use both semantic and co-occurrence scoring, we can extract even more keywords. The proposed ECM method has two main parts, candidate extraction and semantic filtering. This experimental result further proves the validity and utility of our proposed semantic filtering approach.

4.10 **Summary**

There are many researchers who have tried to identify concepts from clinical narratives e.g. (Batool, Khattak, Kim, & Lee, 2013), (Hina, Atwell, & Johnson, 2013), (Gorrell et al., 2016) and (Siddiqi & Sharan, 2015). Also many have attempted to map text to ontology such as UMLS, for example (Batool et al., 2013), Metamap and cTATES tools etc. From the literature review and from our experiments it seems that much more progress is required in this field. Even state-of-the-art tools like cTAKES have produced very low recall with our dataset.

The main contribution of this chapter is to demonstrate that a generic phrase extraction method followed by semantic filtering may perform equally well or better than a model trained with human annotated data. The experimental results highlight that the simple algorithms like n-gram/chunking may work as well as other complex algorithms. Simple n-gram/chunking with semantic filtering may produce better results than the methods that only use statistical measures. This also confirms the conclusion made by Frantzi et al. (2000) that a simple method may perform equally well. The ECM method has been validated with both our own GRiST dataset and the i2b2 dataset. We have also shown that adding semantic phrase ranking improves the performance of RAKE on the semeval2010 phrase extraction dataset.

We are interested to represent each GRiST node not only as a “bag of words” but also as a semantic vector. This could allow us to measure semantic similarity among them, which in turn might reveal important patterns. It might also help to detect the presence of a node in the text by comparing the semantic similarity of a node with the extracted phrases. The next chapter discusses semantic processing of phrases, GRiST nodes and patients in detail.

5 Semantic Processing (Exploratory)

5.1 Introduction

This chapter describes the semantic processing of extracted key phrases, building a semantic profile of a patient, semantic representation of GRiST nodes and mapping GRiST nodes to SNOMED-CT concepts. These were exploratory works carried out to find patterns that might assist us to achieve our main objectives. The activities on this chapter were based around the following hypotheses:

> Hypothesis 1: Word vector can be used as a concept stemming mechanism.
>
> Hypothesis 2: A document vector could be used as a semantic representation of a patient more effectively than other methods.
>
> Hypothesis 3: Representing GRiST nodes semantically may highlight interesting patterns.

The simplest method of representing a patient by using extracted key phrases is to represent the patient as a bag-of-phrases. I have looked for other alternative methods such as document vectors, lists of relational triplets given by OpenIE (Angeli, Johnson Premkumar, & Manning, 2015) system. A recent review of the various linguistic representation methods of semantic content can be found in (Schubert, 2015). Each GRiST node was semantically represented by a vector and interaction among them has been analysed.

Many semantic dictionaries and resources have been developed such as WordNet, VerbNet (Schuler, 2005), and FrameNet (Baker, Fillmore, & Lowe, 1998) to understand the natural language. A review of this method can be found in (Giuglea & Moschitti, 2004). Research has shown that semantic knowledge of the domain helps to improve the knowledge extraction by reducing false positives by up to 75% (Livingston, Johnson, Verspoor and Hunter, 2010). The following sections describe each of the exploratory works with their experimental outputs and critical analysis.

5.2 Dependency Based Similarity

This is an exploratory experiment that I have carried out to find semantically similar words by using Stanford dependency relationships. There are many semantic similarity measures found in the literature and most of them are based on WordNet. WordNet is a manually created similarity database. WordNet based similarity is described in (Gao, Zhang, & Chen, 2015) and in (Pedersen & Michelizzi, 1998). A review of the different WordNet based similarity can be found in (Meng, Huang, & Gu, 2013) and in (Agirre et al., 2009).

Taking the grammatical link between words can improve the result (Padó & Lapata, 2007). Many researchers have taken the grammatical structure into consideration while building a model, a review of these studies can be found in (Padó & Lapata, 2007) and (Turney, 2013). Pado (2007) used graph distance between words and Turney (2013) used subject-verb-object type relationships.

Word2vec algorithm uses surrounding words of a word to create a vector represenatation of a word (Mikolov, Yih, et al., 2013). Cosine similarity distance can be used as a metric to calculate word similarity. I believe dependency relationships can also be used to extract similar words. I have developed an algorithm that can find similar words without using WordNet or Word vector. The algorithm uses Stanford dependency relationships as its attributes.

Firstly, comments were split into sentences then each of the sentences was parsed by the Stanford dependency parser. The XML parse tree was then processed to find dependency tuples. These tuples were then saved in a database table.

Sentence: “Remains verbally hostile and aggressive to staff”

Stanford dependency parse tree:

```
<dependencies type="collapsed-ccprocessed-dependencies">
        <dep type="root">
          <governor idx="0">ROOT</governor>
```

```
      <dependent idx="1">remains</dependent>
    </dep>
    <dep type="advmod">
      <governor idx="3">hostile</governor>
      <dependent idx="2">verbally</dependent>
    </dep>
    <dep type="xcomp">
      <governor idx="1">remains</governor>
      <dependent idx="3">hostile</dependent>
    </dep>
    <dep type="cc">
      <governor idx="3">hostile</governor>
      <dependent idx="4">and</dependent>
    </dep>
    <dep type="xcomp" extra="true">
      <governor idx="1">remains</governor>
      <dependent idx="5">aggressive</dependent>
    </dep>
    <dep type="conj:and">
      <governor idx="3">hostile</governor>
      <dependent idx="5">aggressive</dependent>
    </dep>
    <dep type="case">
      <governor idx="7">staff</governor>
      <dependent idx="6">to</dependent>
    </dep>
    <dep type="nmod:to">
      <governor idx="3">hostile</governor>
      <dependent idx="7">staff</dependent>
    </dep>
  </dependencies>
```

Example of a tuple extracted from the dependency relation shown below:

```
<dep type="advmod">
```

```
<governor idx="3">hostile</governor>
<dependent idx="2">verbally</dependent>
</dep>
```

Table 19 Dependency tuple in table

Type	Governor	Dependent
Advmod	Hostile	verbally

Dependency based Similarity algorithm:

1. Take a word and find all of its governors or dependents for a particular type.
2. Find other words that share the same governor or dependent and count them
3. The words that share maximum dependents or governors with each other are stored as similar words.
4. To improve accuracy, find out if the target word is also a similar word of its own similar words. (This is optional).
5. Output similar words with similarity ranking (here simply match count is used).

The following table illustrates some example words and their semantically similar words:

Table 20 Similarity by dependency and word vector

Word	Similar words by Stanford dependency	Similar words by google news trained word vector	Similar words by PubMed trained word vector
Depression	Mood, thought, problem, health, abuse, this, behaviour, stress, symptom,	bipolar_disorder, depression_anxiety, mental_illness, psychosis, alcoholism, depressive, suicidal_ thoughts, schizophrenia,	Depressive, anxiety, somatization, dysthymia, suicidality, PTSD, psychotic, DSM-defined, mood,

	feeling, pain	anxiety_disorders, psychological_distress	phobia
Misuse	Behaviour, attempt, abuse, alcohol, thought, this, problem, harm	misused, misusing, misuses, misappropriation, abuse, pilferage, Misuse, misspending, misapplication, use	Abuse, illicit, dependence, abusers, drug, addictions, self-medication, marijuana, sedative
Cannabis	Alcohol, drug, substance, heroin, food, amount	marijuana, Cannabis, Marijuana, dagga, heroin, cannabis_resin, ganja, medicinal_marijuana, cocaine, methamphetamine	Marijuana, polysubstance, polydrug, ecstasy, heroin, alcohol, illicit, poly-substance, methamphetamine, hashish
Alcohol	Drug, substance, behaviour, food, cannabis, heroin, misuse, risk	Alcohol, alcoholic_beverages, booze, alcoholic_beverage, alcoholic_drinks, drink, liquor, underage_drinking, drinking, binge_drinking	Cannabis, marijuana, non-alcohol, polysubstance, tobacco, hashish, cigarette, abuse, smoking, crack
Feel	feel, have, be, say, state, report, do, make, take, express,	feeling, felt, feels, think, know, really, Feeling, sense, definitely	Think, forget, felt, want, say, really, something, remember, afraid, know, anything, hear
Need	need, support, care,	needed, want, needs, must, needing, can, should, do, necessary, imperative	Necessity, needs, strive, must, needed, will, should, necessary, continue, future, needing, help, seek,

In the above table, the second column shows the similar words found when using the proposed dependency-based similarity method. The word vector based similar words, shown in the third column, were taken from the web service (http://bionlp-www.utu.fi/wv_demo/), which was trained on the “google world news negative 300” dataset. The fourth column displays similar words based on the word vectors, which were trained on the PubMed dataset (Pyysalo et al., 2012).

This exploratory work shows promising results. For example, it finds similar words for the word ‘alcohol’ such as drug, substance, behaviour, food, cannabis and heroin. It was outside the scope of this research to work on this any further. Dependency based word embedding is described in a recent paper by Levy & Goldberg (2014), which produced a slightly different set of results than a model trained on a “bag of words”. For the purpose of this research we have used a model and algorithm developed by Mikolov, Sutskever, et al. (2013) as described previously in Chapter 4.

5.3 Phrase Reduction

Numbers of phrases extracted by the n-gram method are huge, even after filtering them via. a phraseness filter, as described in the ECM method earlier. It is desirable to reduce this huge number of phrases to a small set that may still reasonably represent the context. Traditional ways of doing this are via character stemming such as the Porter algorithm (Porter, 1980) and/or phrase edit distance such as the Levenshtein distance (Levenshtein, 1966). We may use semantic vector for concept level stemming. The following sections describe both types of stemming and compare their results on GRiST data.

5.3.1 Software Setup

For this research, I have used the word2vec tool developed by Mikolov, Corrado, et al. (2013) and available from (https://github.com/dav/word2vec). To find the vector of a

word a web service has been created. Word vector was extracted based on the google word vector model (GoogleNews-vectors-negative300.bin). An HTTP server program was written in the C programming language with the help of open-source libraries (https://code.google.com/p/word2vec/) and (https://www.cs.utah.edu/~swalton/listings/sockets/programs/part2/chap6/simple-server.c).

Given a word, this server output 300 comma-delimited symantic attributes of the given word. It is very fast and can be used from other programming languages such as Java and PHP. I have looked at a few other, mainly Java based, tools to calculate word vector but ultimately created my own due to the slow processing speed of the other tools.

5.3.2 String Stemming

Phrase extraction algorithms generally produce lots of phrases. The first challenge was to filter them and produce a list of phrases that were actual phrases and not just simply two words appearing together. For this task, the ECM method was used as described in the previous chapter.

The second challenge was to reduce the number of phrases by stemming. Levenshtein distance is a popular method for this. This string similarity technique can filter singular, plural, tenses etc. and different forms of the word to its stem. This phrase stemming process may reduce the number of phrases, but it does not compare similarity based on word semantic. Nonetheless, we can use string stemming to reduce the number of phrases.

There are many classical stemming algorithms available such as Lovins, Dawson, Porter, Paice/Husk, Krovetz (Moral, de Antonio, Imbert, & Ramírez, 2014). Comparison of them can be found in (D. Sharma & Cse, 2012). For this research, I have used mainly Levenshtein distance which is explained in (Haldar & Mukhopadhyay, 2011). This performed better than all others that have been tried with a sample dataset. In this document, the terms ‘string stemming’ or ‘syntactic stemming’ refers to the reduction of phrases by Levenshtein distance measure.

There is a Java library called simmetrics, which was developed by Sam Chapm of Sheffield University, UK and has been released as an Open Source library. It is accessible from (https://github.com/Simmetrics/simmetrics). This is a Java library of similarity and distance metrics e.g. Levenshtein distance. It has implemented many algorithms and a complete list of them can be found in its Github repository source code.

The measure metrics are normalised in the simmetrics library, so the maximum similarity is 1 and no similarity is zero. This allows us to define how much similarity we want, for example 0.80 or 0.90 etc. The level of similarity is a parameter of the algorithm so may differ from dataset to dataset, or on requirements. The normalisation also helps us to compare this similarity with the similarity produced by word vector (cosine similarity).

5.3.3 Semantic Stemming

String similarity measures such as Levenshtein distance does not work based on the actual meaning of the words. To reduce the number of phrases semantically we may use cosine similarity between two phrase vectors. We can use the following simple steps to accomplish concept stemming.

Step 1: Take a word and find its word vector
Step 2: Match the word with each of the words in the concept list
Step 3: If a match is found within a threshold value then use the word already in the concept list. Or if a match is not found then add this concept word to the concept list.

By applying the above simple steps, we can extract a reduced set of concept words that represent the domain concepts. It may be better to use the average of the vectors of all the words that match with a specific concept word to represent the vector of that concept.

This technique has many potential benefits. If we compare the nodes based on string matching this would not cover the words that are different by character comparison but similar semantically. For example, the words ‘anxiety’ and ‘paranoid’ are different in spelling but semantically they are close. The application of word vectors and cosine similarity can aid us in this context.

I have used the simmetrics library for string based similarity measures and cosine similarity for vector base similarity measures. The idea was to compare these two methods and gain an insight into their potential future use.

5.3.4 Experimental Results

For this exploratory work, I have used bigram phrases only. A Java program was developed to implement the phrase compression methods by both string stemming and semantic stemming as described earlier. The software was run with different levels of matching threshold. The following are the results of the experiments.

Total number of phrase = 3185

Table 21 Phrase reduction results

Similarity	String stemming number of phrase	Semantic stemming number of phrase
0.99	3185	2924
0.90	2827	2861
0.80	2541	2434
0.70	2183	1673
0.60	1355	1011

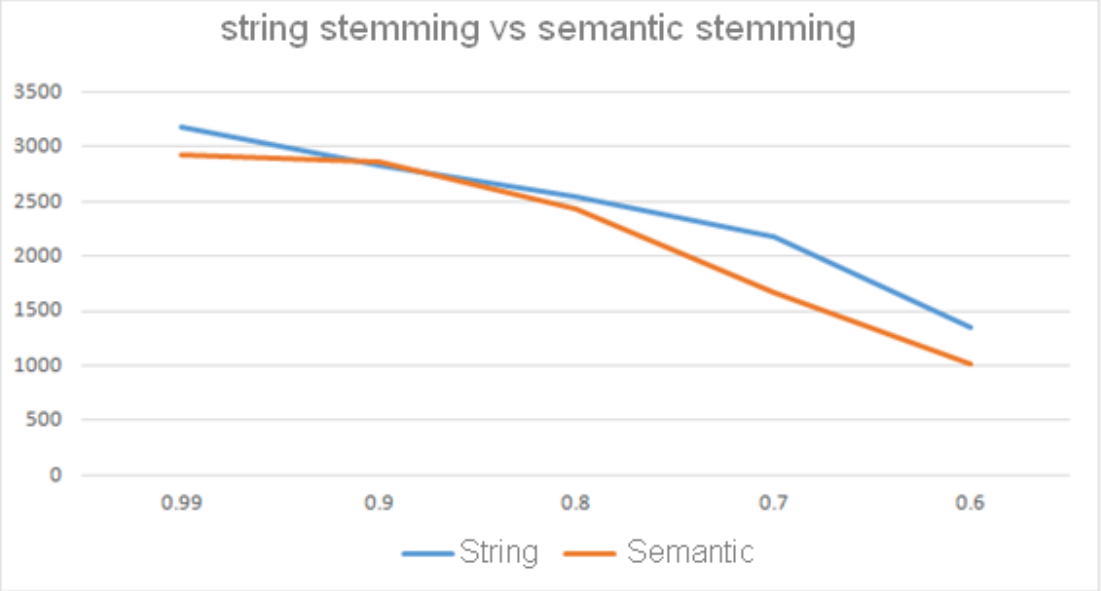

Figure 11 string vs semantic stemming results

From the above chart, we can see that semantic compression has more impact. At 0.70 similarity phrase compression=1-2183/3185=0.32 (approximately 32%) and vector based compression is 48%. More compression does not mean more quality. It has been seen from the data that after less than a 0.70 similarity measure the phrases start to differentiate semantically. From our dataset, we have seen that about 0.80 similarity measure gives better results.

Experimental results show that we can use string based similarity to do the stemming and after that, we can perform a vector based similarity to reduce the number of extracted phrases semantically. Some sample phrases are shown in the table below.

Table 22 Phrase stemming by string match and vector similarity

Phrase	Phrase concept	Vector concept
alcohol abuse	alcohol abuse	Misuse alcohol
occasional suicidal	occasional suicidal	frequent suicidal
anger issue	anger issue	anger problem
alcohol intoxication	alcohol intoxication	drinking alcohol
anger outbust	anger outbust	anger toward

From the above sample, we can see “anger issue” and “anger problem” both considered as the same concept based on the semantic distance of the word vectors. Levenshtein distance would not work in this case, as the similarity would have been low. This result shows support towards the first hypothesis in this chapter that the vector based similarity measure can be used for concept stemming. One must find the suitable value for the acceptable similarity measure by trial and error. We have observed that about a 0.80 value for similarity provides semantically similar results. Please note 1 means complete match and 0 means no match at all by cosine similarity measure.

5.4 Semantic Profile Representation

Developing a semantic profile of a patient based on the comments is desirable. We can assume that in real life when a clinician assesses a patient they create a profile of that patient in their mind based on the data available and the physical observations. Some of the methods that have been considered for building a patient profile are described below.

5.4.1 Bag of Concepts

We could simply use the extracted concepts from a patient description to represent the patient. We can create a list of all possible phrases and create a vector with value 0 for the non-existence of the phrase and 1 or actual count as the existence of the phrase. The resulting dataset can then be used in machine learning tools like Weka. Many text classification algorithms can create the bag-of-words themselves to run the classification task on raw text. We aimed to explore other possible options and put more emphasis on semantic rather than syntactic similarity.

5.4.2 OpenIE tuples

This is quite similar to the “bag of concepts” method described previously but uses OpenIE extracted relations instead. Relation triples produced by the Open Information Extraction (OpenIE) system are useful for question answering, inference, and other IE tasks (Angeli et al., 2015).

> *Open information extraction (open IE) refers to the extraction of relation tuples, typically binary relations, from plain text. The central difference is that the schema for these relations does not need to be specified in advance; typically, the relation name is just the text linking two arguments. For example, Barack Obama was born in Hawaii would create a triple (Barack Obama; was born in; Hawaii), corresponding to the open domain relation was-born-in (Barack-Obama, Hawaii).*
> *(*Source: http://nlp.stanford.edu/software/openie.html)

For sentence “Born in Honolulu, Hawaii, Obama is a US Citizen.”

We get the following triples:

Table 23 OpenIE tuples

Entity	Relation	Object
Obama	Is	US citizen
Obama	born in	Honolulu, Hawaii

From the above table, we can see that if we extract tuples from each of the sentences from patient notes than we can build a list of tuples. We can then compare which patient has similar types of relationships and whether they can be used to represent the patient and ultimately predict suicide. For this purpose, the OpenIE system was downloaded from (http://nlp.stanford.edu/software/openie.html) and a tuple generation web service was created in which we can feed a sentence and get all the tuples as return values.

The following are some typical outputs from the OpenIE system:

Sentence: Admits to attacking a man with a cricket bat on the day she was arrested.

Table 24 OpenIE parsed results

Probability	Entity	Relation	Object
0.42	She	be arrest with	cricket bat
0.42	She	Be	with cricket bat on day arrest
0.42	She	be arrest on	Day
1	Admit	Attack	Man
0.42	She	Be	Arrest

From the above we can see that the phrases “attack man” and “be arrest” could be potential indicators of harm to others (hto) and could identify an adverse situation.

Sentence: Previous reports of setting fire to his clothing (jeans) when in prison after obtaining a lighter from another prisoner.

Table 25 OpenIE parse results of the second sentence

Probability	Entity	Relation	Object
1	Report	set fire when to	he clothing
1	He	Clothing	jeans
1	Report	set fire to	he clothing
1	previous report	set fire when to	he clothing
1	previous report	set when	fire
1	previous report	set fire to	he clothing
1	Report	Set	fire
1	previous report	Set	fire
1	Report	set when	fire

From the above table, we can see that the “set fire” could be a potential indicator of harm to others.

It appears that we may build a list of tuples and classify the patient based on the tuples presence. We can ignore the first entity column, which most likely includes a patient's name. But we face a serious challenge here. In the first example, we have 5 tuples and in the second example, we have 9 tuples. Not all of these tuples are relevant, and we observe a huge number of them for each sentence and ultimately for each patient.

There needs to be a semantic processing to filter these tuples before they can be used for profile building. Some of the simple methods like frequency count have been used for this purpose without success. It appears that we need more semantic analytical power than just simple occurrence counting.

As a demonstration of the usefulness of the Phraseness filtering by ECM method described previously, we can apply it to the extracted tuples. Approximately 100,000 sentences were first analysed by the OpenIE tool and then the third part of the tuples was considered as a phrase. The following tables show lists of phrases with and without the phraseness algorithm. The number after the equals sign is the occurrence count.

Table 26 Phraseness algorithm applied on OpenIE extracted data

Without phraseness filtering	With phraseness filtering
the past=61(frequency)	suicidal thought=32
suicidal thought=32	protective factor=20
self harm=30	suicidal ideation=15
he wife=28	mental health=14
he life=27	suicide attempt=14
he mother=23	mental health service=10
long history=21	fleeting thought=9
protective factor=20	physical health=8
she life=20	hospital admission=8
he family=18	verbally abusive=7
risk of suicide=16	delusional belief=6
thought of suicide=16	sexually abuse=6
she mother=16	illicit substance=6
she husband=15	sleep pattern=6
suicidal ideation=15	family member=6

mental health=14	health service=6
she home=14	memory problem=6
suicide attempt=14	fellow patient=5
she medication=13	serious attempt=5
she child=11	could vulnerable=5

We can see that the data generated by sophisticated tools like OpenIE still needs filtering. Arguably, to limit the complexity one can simply generate n-grams from raw text and apply filtering algorithms. In the case of 'bag of concepts' or 'bag of triples' the resulting number of total attributes for classification can be huge. Representing patients with semantic vector is considered more manageable and it has been shown that the vector representation performs better (Le & Mikolov, 2014). We may also simply use raw text to represent patients and use text classifiers. Some of the classification experiments and results are shown in Chapter 6.

5.4.3 Document Vector

Document Vector is a vector representation of a document built by combining the word vector of its constituent words. Creating a document vector and the use of it for text classification has achieved state of the art results as claimed by Le & Mikolov (2014). They argue that the bag-of-words features have two major weaknesses: they lose the ordering of the words and they also ignore semantics of the words. Empirical results show that document vector outperforms bag-of-words models as well as other techniques for text representations (Le & Mikolov, 2014).

In the GRiST system, we have comments attached to the ontology nodes and if we concatenate all these comments then we can represent each assessment by this resulting text. We may apply the document vector idea and create a semantic representation of an assessment by creating a vector from this text.

There are many methods for calculating document vector from the word vectors. The simplest one is taking the average of the entire constituent words vector. For this research, I have used an implementation of the algorithm as proposed by Le & Mikolov (2014). A Python based implementation of the algorithm is available from (https://github.com/klb3713/sentence2vec). The created document vectors were used to classify risk level, which is described in chapter 6.

5.4.4 Profile Representation Summary

From the above mentioned methods we can observe that “bag of words” and “openIE tuples” both require filtering. For any subsequent application of the created profile we would also need string matching. This makes them complex and unsuitable for machine learning.

In comparison to other methods, the document vector approach is more flexible, and this highlights our second hypothesis of this chapter. We may represent a patient by a vector calculated from clinical narratives. A detailed description of the process of creating the document vector and its use in classification is given in Chapter 6 where it has also been compared with other classification methods.

5.5 Semantic Representation of GRiST Nodes

After considering the semantic representation of phrases and patients, the semantic representation of the GRiST nodes has been explored. GRiST ontology has leaf nodes, which are often linked to a question for gathering information about a patient. There are two types of answers - one is a numeric value and the other is an optional comment. All the comments are stored in a database corresponding to its GRiST node name. To build a semantic representation of a node the stored comments were used.

Once we construct a semantic representation of a node, we may use it to better understand the data and improve the GRiST system. This might allow us to extract phrases from comments and attach them automatically to a node based on the cosine similarity measure. This technique may also help us to determine a numeric value of a node for a given patient from the textual comments.

Several experiments have been conducted to explore and understand the dataset and find interaction patterns in the GRiST ontology nodes. These experiments prove the third hypothesis of this chapter, that representing GRiST nodes semantically may highlight interesting patterns. These experiments are described below in their own separate sections.

5.5.1 Experiment 1: Finding Relevant Phrases within a Node

Generating a list of relevant phrases for a node is time-consuming. The method described here shows how we can do this automatically. Bigram phrases have been extracted for each node and stored in a database table. There were many phrases in a single node and finding which phrases were relevant to the node was a challenge. Understandably, there was some overlap of comments and the same phrases may appear in more than one node. To overcome this problem a method was applied based on the following algorithmic steps.

Automatic generation of relevant words:

1. Calculate vector for each phrase.
2. For each phrase in a node calculate its distance with all other phrases in the same node and average the distances.
3. Sort the phrase based on the average distance value. A phrase that has a high value means they are more closely related to this node.
4. Do this for every node.

The core assumption was that a relevant phrase would have many similar phrases of itself in the same node. Therefore, if we measure similarity between each phrase and

add the scores then the highest scoring phrase is the most relevant to that specific node. No other filtering has been done and the algorithm was run on the extracted phrases. The following are the results for some of the nodes.

Table 27 Relevant phrase within a node

Node Name	Phrase=score
Suic (suicide)	sucidal thoughts=0.78 serious attempt=0.72 expressing suicidal=0.69
suic_curr_int (suicide current intention)	committing suicide=0.61 harm suicide=0.50 longer experiencing=0.472 longer feels=0.46 experiencing suicidal=0.39 feels positive=0.30
suic_pot_trig (suicide potential trigger)	feels isolated=0.37 feels lonely=0.37
suic_ideation (suicide ideation)	having thoughts=0.51 having fleeting=0.49 denies thoughts=0.28
Sh (self-harm)	intrusive thoughts=0.56 self injury=0.53 harming behaviours=0.52 harm attempts=0.49 superficial cutting=0.46 coping strategy=0.33 coping mechanism=0.33
Hto (harm to other)	physical aggression=0.57 aggression towards=0.55 damaging property=0.53 aggressive towards=0.51 verbal aggression=0.50 physically aggressive=0.43 damage property=0.40

We can see from the above table that node meaning and highly scored phrases closely resemble one another. We may build a semantic vector for each node based on these phrases. Any new phrase can be matched against the node semantic and the closer the match the more likely that phrase would relate to that node.

When we do not have manually annotated phrases then we can use this method to extract meaningful phrases for a particular node. Once we have relevant phrases then we can use them to build the vector representation of the node. Starting from nothing this technique can build better semantics for each node as more and more data becomes available.

This technique has real life application, as mapping extracted phrases to a specific node would indicate activities related to that node. For example, “verbally aggressive” would automatically match with node hto (harm to others).

5.5.2 Experiment 2: Clustering Phrases by Semantic Vector

The GRiST ontology nodes ideally should capture all possible aspects of suicide risk factors. For example, they could be financial problems, health problems and relationship problems etc. We expect an ontology should cover all the aspects of the relevant domain. I have assumed that we may be able to cluster the phrases based on its vector value and find clusters that may represent the different aspects of the domain.

Before clustering the phrases, we need to find the vector representation of the phrases. For multi-word phrases, the phrase vector was calculated by averaging its constituent word vector. Each of the words was passed to the web service to obtain the word vector and then it was used by the java program. For cluster analysis, SimpleKMeans algorithm from Weka Machine learning package (Hall et al., 2009) was used to create clusters of different sizes.

Table 28 Phrase clustering 20 clusters

Cluster 0	Cluster 1	Cluster 2	Cluster 3
information sharing, problems reading, counseling services, personal hygiene, community living, anti social, behavioral changes,	black head, hand lacerations, arm bruises, big head, head butt, head injuries, banging head, toes claw, head banging,	interpersonal skills, coping skills, interpersonal relationship, learning skills, social skills, selfcare skills, understanding behaviour, literacy skills,	serious injury, handsuperficial injury, blister finger, hip replacement, fractured jaw, ankle injury, neck spondylosis, broken wrists, stomach operation, abdomen wound,

Table 29 Phrase clustering 50 clusters

Cluster 0	Cluster 1	Cluster 2	Cluster 3
personal hygiene, anti social, social activities, spatial awareness, social anxiety, social skills, phobia social, social event, lifestyle education, social services, increasing awareness, personal awareness, social drinking, employment education,	black head, hand lacerations, arm bruises, big head, head butt, head injuries, banging head, toes claw, head banging,	interpersonal skills, coping skills, interpersonal relationship, learning skills, selfcare skills, understanding behaviour, literacy skills, verbal communication,	serious injury, hand injury, blister finger, hip replacement, fractured jaw, ankle injury, neck spondylosis, abdomen wound, arm fractures, minor injury, leg dvt, groin abscess, sliding hernia, recent injury, fracture femur, tendon repair, broken rib

The analysis of the created clusters shows that even though there are lots of similarities exist, but anomalies are also visible. For example, in the above Table 29 we can see ‘head injuries’ in cluster 1 and ‘recent injury’ in cluster 4. Many different types of

concepts are mixed in the same cluster. Even though there is lots of accurate clustering, it is still challenging to create a clear semantic boundary. It could be that words are spread over a huge number of contexts and using cluster sizes 20 or 50 does not capture the correct representation.

My initial thought was to cluster the words by word vectors, then connect each of the clusters to a specific GRiST ontology node. From the above analysis, we see that getting a semantically meaningful cluster is challenging. Alternatively, I have tried to connect phrases to the GRiST node semantically. The detail of that process is described in the following section.

5.5.3 Experiment 3: Finding Phrases Similar to a Node

Semantic vector representation of the GRiST nodes was calculated. From the database, I have extracted the words that appear in a specific node. Then word vector for each of these words was calculated. The word vector was based on the google-word-vector model as described previously. I have used only the manually annotated words for this purpose. The node semantic vector was calculated by averaging the semantic vector of each of the occurring words.

In the next step, I have queried every single word from the database. Then word vector was calculated for the selected word and then the cosine similarity between a word and a node was calculated. For each node, I have listed the top 10 words that match more with the node. Some of the results are shown in the table below.

Node and Words Semantic Matching Method:

Step1: Query all nodes from the database
Step2: For each node find relevant words in the n-gram table
Step3: Calculate word vector for each word
Step4: Average the words vector to get the node vector
Step5: Select a node

Step6: Extract words from n-gram list and calculate word vector

Step7: Calculate the cosine similarity between word and the node vector

Step8: Sort the word by its similarity to the node

Step9: Display the top ten words per node

Table 30 Phrases similar to a node

Node name	Top matching words
suic_discovery	suicidal=0.63 overdose=0.59 unprescribed=0.59 medication=0.59 alochol=0.57 overdosed=0.56 psychiatric=0.56 methadone=0.55 overdoses=0.55 overdosing=0.54 temazepam=0.54
suic_lethality	unprescribed=0.69 methadone=0.68 medication=0.68 painkillers=0.67 overdose=0.66 drugs=0.66 temazepam=0.65 alcohol=0.64 medications=0.64 overdosing=0.63 antidepressants=0.63
suic_regret	embarrassed=0.66 despondent=0.64 angry=0.63 ashamed=0.62 unhappy=0.61 sorry=0.61 remorseful=0.61 suicidal=0.61 scared=0.60 frustrated=0.59 regretful=0.58
suic_leth_insght (suicide lethality insiht)	unprescribed=0.62 suicidal=0.60 medication=0.60 overdosing=0.58 drugs=0.58 disinhibited=0.58 overdose=0.57

Node name	Top matching words
	methadone=0.56 psychosis=0.56 antidepression=0.56 overdoses=0.56
suic_prosp_leth (potential lethality of prospective suicide method)	overdoses=0.66 overdose=0.66 methadone=0.65 drugs=0.64 medication=0.64 painkillers=0.63 unprescribed=0.62 overdosing=0.61 antidepressants=0.60 medications=0.60 methodone=0.59
suic_eol_prep (suicide end of life preparation)	suicidal=0.56 concerned=0.56 worried=0.55 concern=0.52 anxiety=0.50 afraid=0.50 distress=0.48 paranoid=0.48 concerns=0.48 scared=0.48 harm=0.47
gen_sh_cuts (general self-harm cutting)	abdomen=0.63 wrists=0.62 throat=0.61 wounds=0.57 bruises=0.54 knife=0.49 machette=0.46 abrasion=0.45 selfdefence=0.44 arms=0.44 abcesses=0.44

From the above empirical data, we see that ‘suic_discovery’ and ‘suic_lethality’ share many relevant phrases. For both ‘suicide discovery’ and ‘lethality’ contexts we generally expected to see words like "overdose" and 'unprescribed' drugs.

In the node 'suic_regret' we see the words 'embarrassed', ‘ashamed’ and 'remorseful'. Again, in this context, we probably expected to see if somebody attempted suicide and failed to do so.

The ‘suic_eol_prep’ (end of life preparation) has the words like ‘concerned’, ‘worried’, ‘anxiety’ and ‘paranoid’. Again, this seems similar to the context.

The node ‘suic_p_trig_mtch’ (potential trigger) contains the words ‘anxiety’, ‘depression’, ‘traumatic’ and ‘paranoid’.

The node ‘gen_sh_cuts’ (self-harm cuts) contains the words ‘wrists’, ’throat’, ‘wounds’ and ‘knife’.

The node ‘hto_dest_prprty’ (harm to other destroy) contains the words ‘assaulted’, ‘stabbed’, ‘violent’ and ‘altercation’.

It is clear from the above example that we can attach relevant words to a specific ontology node by the proposed method. The semantic vector of an ontology node attracts the relevant words. We do not have to annotate a big list of words. If we annotate a few words and build the semantic representation of a GRiST ontology node from them, we can then automatically extract other words that may be relevant to the context of the node.

Some of the nodes for example, ‘suic_eol_prep’, ‘suic_leth_insght’, ‘suic_discovery’ and ‘suic_lethality’ share many common words. I would argue that in all of the cases they are very closely related nodes. If we look at the GRiST structure, we see that some nodes have only subtle difference. We expect ‘suic_discovery’ and ‘suic_lethality’ to be related to actual suicide attempt or process. This leads us to investigate how the GRiST nodes are themselves semantically related, which I have done later in this chapter.

We can see that if we want to develop an ontology from raw data, using clustering based on the word vectors, it might prove more challenging. However, the other way around, when we have an ontology, then we can do semantic analysis and potentially improve the ontology easily. The next experiment looked at how GRiST nodes are themselves clustered based on their vector representation. This could indicate which nodes are similar.

5.5.4 Experiment 4: Clustering Nodes by Vector

GRiST ontology has a huge number of nodes - a total of 446. The semantic vector of a node represents it semantically. The objective of this experiment was to cluster the nodes based on its semantic vector value and to find out which nodes are closely related with each other. As GRiST has many nodes, this might provide us a way to review the dimension of the ontology. We could try with other established methods such as principal component analysis, but doing this task using comments and word vectors could be advantageous, especially when numerical data may not be available.

I have calculated the node vectors by averaging the word vectors of the frequent words, which appeared in the nodes. A list of words was queried from the database for a particular node and then word vectors for the words were calculated by using the word vector HTTP server as mentioned previously. Then the average of the word vectors was chosen as the node vector. Once I have the vector for all the nodes, then they were clustered with the SimpleKmeans clustering algorithm by using Weka library. I have written all the relevant code to build a Java based tool that does this task efficiently.

The following tables show the clustering of the nodes. The number of cluster is a variable of the software tool. I have carried out analysis with many number of cluster sizes. Some of the clusters with a total cluster number of 10 and 20 are shown in the tables below.

Table 31 GRiST node clustering 10 clusters

Cluster 0	Cluster 1	Cluster 2	Cluster 3
suic_leth_insght, suic_lethality, gen_drug_misuse, gen_alc_misuse, gen_diet_weigt_chg, gen_meds_concord, gen_rsk_behavr, gen_unint_risk_behavr, suic_prosp_leth,	gen_low_mood,	phys_vuln, gen_hostile, gen_prob_act_par_del, hto_steps_plan, gen_empathy_abil, gen_threat_move, gen_detached, hto_hi_rsk_ideatn, gen_risk_aggrsv, gen_chall_bhvr,	gen_learn_disab, gen_decision, gen_com_imp, gen_diet_eating, gen_phys_hlth_disa, sn_hygiene, gen_phys_hlth_det, gen_cog_think_mem, gen_app_diet,

		gen_voice_dang_o, gen_voice_dang_s, gen_gut_assmnt,	sn_skin, sh_lethality_mth, gen_phys_hlth_pain,

Please note: The meaning of the GRiST node can be found in Appendix A.

Table 32 GRiST node clustering 20 clusters

Cluster 2	Cluster 3	Cluster 4	Cluster 6
gen_detached, gen_voice_dang_o, gen_voice_dang_s, hto_strgth_ideatn, gen_congruence,	sn_recnt_app_chnge, sn_hair_clothes, sn_hygiene, gen_liv_skills, sn_skin,	gen_distrss_b_lang, gen_anx_emotns, gen_mood_swings, gen_avoid_eye_conta ct, gen_mania, gen_angry_emotns, gen_sad,	gen_alc_misuse, suic_planning, gen_meds_concord, sn, sh, gen_mentl_insght, family_ment_hlth, gen_nd_hlp_diff,

Please note: The meaning of the GRiST node can be found in Appendix A.

Finding clear patterns from these clusters is challenging. Clearly some of the conceptually similar nodes appear in the same cluster. For example, in cluster 4 we see distress, anxiety, mood swings and mania, which are semantically similar nodes. In cluster 3, we have nodes like hygiene, self-neglect etc. This shows that there is some semantic coherence among the nodes, which appear within the same cluster. But at the same time, cluster 6 contains the nodes such as alcohol misuse, suicide planning, self-harm and family mental health. These nodes represent semantically different concepts, but they have all appeared in the same cluster. This demonstrates that only some of the clusters represent semantically coherent themes.

GRiST has a huge number of nodes and sometimes the differences between them are very subtle. For example, ‘self neglect hygiene’ and ‘self neglect skin’ are very similar concepts. The results of this experiment may be used for exploration purposes and to gain a better understanding of the GRiST ontology. We may also explore the node relationships simply by measuring the cosine similarity among them. This is discussed in the following section.

5.5.5 Experiment 5: Inter Node Cosine Similarity

In the previous sections, I have discussed the semantic representation of GRiST nodes. Once we have calculated the semantic vector of the GRiST nodes we can compare how close they are from each other. Cosine similarity was used to calculate the similarity between the nodes.

Each of the GRiST nodes was compared against other nodes and the top ten closest nodes were extracted. In the cosine similarity measure score 1 means very similar and score 0 means not similar. The following table shows a partial list of nodes and their top ten similar nodes. A complete list can be found in Appendix B.

Table 33 GRiST internode node similarity

Node Name	Cosine Similarity
gen_hostile	gen_hostile=1.00 gen_chall_bhvr=0.91 hto_pot_trig_mtch=0.90 hto_pot_trig=0.90 gen_neigbrhd_rsky=0.89 gen_angry_emotns=0.89 risk_dep=0.89 gen_prob_act_par_del=0.89 gen_unusl_rec_bhvr=0.88 gen_gut_assmnt=0.87 phys_vuln=0.87
gen_mood_swings	gen_mood_swings=1.00 gen_anx_emotns=0.97 suic_pot_trig=0.96 hto_pot_trig=0.96 gen_mania=0.95 gen_unusl_rec_bhvr=0.94 vuln_su=0.94 gen_sad=0.94 gen_angry_emotns=0.94 suic_p_trig_mtch=0.93 sh_pot_triggs=0.93
gen_motivation	gen_motivation=1.00 gen_day_struct=0.92 gen_med_perc_benft=0.91 gen_phys_withd=0.91 gen_diet_eating=0.91 gen_liv_skills=0.90 sn_hair_clothes=0.90

Node Name	Cosine Similarity
	sh_pot_trigs_mtch=0.90 gen_app_diet=0.89 gen_insght_behvr=0.89 gen_mental_withd=0.89
suic_lethality	suic_lethality=1.00 gen_drug_misuse=0.93 gen_unint_risk_behavr=0.92 gen_alc_misuse=0.92 suic_leth_insght=0.91 gen_meds_concord=0.90 gen_rsk_behavr=0.90 suic_discovery=0.90 sh_lethality_mth=0.89 suic_planning=0.89 gen_impulse=0.89
gen_distress	gen_distress=1.00 suic_p_trig_mtch=0.96 gen_sad=0.96 gen_anx_emotns=0.95 gen_life_not_livng=0.94 hto_pot_trig=0.94 sh_pot_triggs=0.94 gen_mood_swings=0.93 suic_id_strngth=0.93 gen_angry_emotns=0.93 emot_vuln=0.93

Please note: The meaning of the GRiST node can be found in Appendix A.

From the above data, again we can see that nodes that are similar by cosine similarity measures seem to be closely related semantically. For example, gen_distress node is closely similar to gen_sad, gen_anx_emotns, gen_mood_swings and emot_vuln. There are some dissimilar nodes such as suic_lethality and gen_drug_misuse, which appear to be similar based on this analysis. While there are some exceptions, the overall pattern of similarity is visible in majority of the nodes.

I have compared this data with the correlation between nodes based on the inputted numerical value. In Appendix B I have included more data. For discussion purposes, data for only two nodes are shown here.

Table 34 Node to node correlation

Node name	Correlation	Cosine Similarity
suic_lethality	suic_lethality=1.00, suic_ser_succd=0.65, **suic_planning**=0.48, **suic_discovery**=0.48, **sh_lethality_mth**=0.40, suic_id_hi_risk=0.38, suic_id_strngth=0.38, suic_id_control=0.34, suic_prosp_leth=0.32, hto_fire_setting=0.27	suic_lethality=1.00 gen_drug_misuse=0.93 gen_unint_risk_behavr=0.92 gen_alc_misuse=0.92 suic_leth_insght=0.91 gen_meds_concord=0.90 gen_rsk_behavr=0.90 suic_discovery=0.90 sh_lethality_mth=0.89 suic_planning=0.89 gen_impulse=0.89
gen_hostile	gen_hostile=1.00, gen_risk_aggrsv=0.71, **gen_angry_emotns**=0.66, gen_threat_move=0.65, **gen_chall_bhvr**=0.60, gen_empathy_abil=0.48, hto=0.48, gen_mania=0.45, gen_reliable=0.45, hto_curr_persp_ep=0.43	gen_hostile=1.00 gen_chall_bhvr=0.91 hto_pot_trig_mtch=0.90 hto_pot_trig=0.90 gen_neigbrhd_rsky=0.89 gen_angry_emotns=0.89 risk_dep=0.89 gen_prob_act_par_del=0.89 gen_unusl_rec_bhvr=0.88 gen_gut_assmnt=0.87 phys_vuln=0.87

Please note: The meaning of the GRiST node can be found in Appendix A.

We can observe that if we only take 10 nodes that are close by similarity or correlation we see some overlap. The vector based similarity is matching with a few numbers of relations found by inter node correlation. There are a few limitations that may be affecting the results.

Firstly, GRiST nodes are too granular in nature. For example, gen_hostile, gen_chal_bhvr and gen_angry_emotns all are very similar concepts. Someone is ‘angry’ may easily imply that he or she is also ‘hostile’ or vice versa. It is possible that clinicians are inputting similar comments in these nodes. I have observed in the data that sometimes clinicians inputted exactly the same text in these fields.

Secondly, text inputting is optional and in fact, numerical inputting is also optional in cases of rapid assessment. The original design of GRiST mainly focused on numeric

data. This causes lots of missing comments in the node. That ultimately might affect this analysis.

Thirdly, in natural language for describing somebody as aggressive or angry someone may use similar words. Capturing subtle differences is very challenging.

The results of this and the previous experiments clearly show a positive trend of similarity between semantically similar nodes even though results are not as accurate as expected. This could improve the understanding of the underlying knowledge structure of the GRIST ontology.

5.5.6 Node Representation Summary

The above experimental results show that representing a GRiST node by semantic vector can be useful. Once a semantic vector of a node is created, we can automatically assign to it extracted phrases from comments. If a phrase has cosine similarity of more than a certain value, then it can be considered very close to that node. We can also compare how close a node is to each other. This technique opens up many possibilities including finding similar nodes and possibly merging them together.

In SNOMED-CT, there is a list of representative phrases for each node. GRiST ontology does not have that. By the techniques described above, we can automatically build a set of phrases that may represent GRiST nodes semantically. This exploratory work could be considered as a contribution towards future research on the GRIST project.

5.6 GRiST and SNOMED-CT (Exploratory)

Ontologies are being used as knowledge structures by many medical expert systems and these ontologies vary in terms of their coverage, completeness and purpose (Cardillo, 2015). Semantic interrelationships among these ontologies are required for

interoperability of clinical systems. To facilitate interoperability with other systems GRiST ontology nodes were mapped to the SNOMED-CT terminology. SNOMED-CT is a standard terminology database and widely used so linking GRiST with SNOMED-CT may facilitate new knowledge discovery.

There has been a lot of research to collect concepts from text and map them to the terminology database like SNOMED-CT, some of them are described below. Matching GRiST nodes to standard terminology like SNOMED-CT may provide the semantic meaning of the node as well as it may be possible to link them to other ontology in the same domain. This was a high level exploration work to see whether it was worth exploring in more detail.

Concepts extracted without focusing on a specific domain may generate lots of phrases which are generally scattered over a large concept space and hardly useful for practical purpose (Hovy, Kozareva and Riloff, 2009). The basic model of mapping is to chunk text to phrases and then map them to the concepts in the ontology (Ducatel, Cui and Azvine, 2006). A review of the recent methods of ontology mapping techniques can be found in (Kaza & Chen, 2008) and (Ramar & Gurunathan, 2016).

SNOMED-CT is a comprehensive reference terminology that allows healthcare providers to record clinical events accurately and unambiguously (Lee et al., 2010). The Unified Medical Language System (UMLS) is a collection of many biomedical vocabularies, which includes SNOMED-CT. In this report, the term UMLS and SNOMED-CT are sometimes used synonymously.

The International Health Terminology Standards Development Organisation is in charge of SNOMED-CT and their website (http://www.ihtsdo.org/snomed-ct) describes the benefits of SNOMED-CT as follows:

- It is a comprehensive multilingual terminology database
- Contents are scientifically validated.
- Is consistent and electronically processable.
- Is mapped to other international standards.
- Is already used in more than fifty countries.

There are researches that have specifically tried to identify medical concepts from free text and map them to SNOMED-CT. For example, an automated system for the conversion of clinical notes into SNOMED clinical terminology proposed by Patrick, Wang, & Budd (2006). Learning formal definitions of terms in the text from SNOMED-CT by Ma & Distel (2013) and automatically mapping concepts in the patients discharge summary to SNOMED-CT by Batool et al. (2013).

Adlassnig (2009) extracted morphemes from clinical texts and mapped them onto concepts from SNOMED-CT. Bleik, Xiong, Yiran Wang and Song (2010) represented full-text documents as a graph using LingPipe's NER concept nodes and relation edges. They mapped extracted terms to the concepts from the UMLS database.

The SyMSS (Syntax-based Measure for Semantic Similarity) system compares semantic similarity between short texts and sentences by taking into account semantic and syntactic information. Psychological plausibility was added to the system by using the previous findings about how humans weigh different syntactic roles, when computing semantic similarity (Oliva, Serrano, del Castillo and Iglesias, 2011).

Pakhomov, Buntrock & Duffy (2005) developed a high throughput modularised system for text analysis and information retrieval that identifies clinically relevant entities in the clinical notes and maps them to the several standardised nomenclatures such as SNOMED-CT.

In this research, I have tried to use the existing tools like cTAKES to extract concepts from the clinicians' comments and used their mappings to SNOMECT-CT terms. The objective was to find out how the concepts in GRiST nodes link to the concepts in SNOMED-CT. This may allow us to discover some patterns, which may eventually help us to find further patterns in the GRIST nodes inter-relationships. Before describing the methods, a brief description of the SNOMED_CT is provided next.

5.6.1 Structure of SNOMED-CT

SNOMED-CT is a core clinical healthcare terminology that contains concepts with unique meanings and formal logic based definitions, organized into hierarchies (IHTSDO, 2014). SNOMED-CT Starter Guide (IHTSDO, 2014) describes that the SNOMED-CT contents are represented using three types of components:

- Concepts: represents clinical meanings and are organized into hierarchies.
- Descriptions: link appropriate human readable terms to concepts.
- Relationships: link each concept to other related concepts.

These components are enhanced by Reference Sets, which facilitate addition of the additional features and enable configuration of the terminology to address different requirements. The following short descriptions are summarised from (IHTSDO, 2014).

Concepts: SNOMED-CT concepts represent clinical thoughts and is organised in hierarchy from general to more detailed. This allows detailed clinical data to be recorded and later retrieved or aggregated at a more general level.

Descriptions: SNOMED-CT descriptions are generic descriptions of a concept. A concept can be described in different ways, each representing a synonym that describes the same clinical concept. Each translation of SNOMED CT includes more language specific description of the concepts. Every description has a unique numeric description identifier.

Relationships: SNOMED-CT relationships link one concept to others semantically. These relationships depict the properties of the concept. One type of relationship is the |is a| relationship, which relates a concept to more general concepts. This |is a| relationship defines the hierarchy of SNOMED-CT concepts.

Reference sets: The Reference sets (Refsets) are used for customisation of SNOMED-CT and it is a flexible approach for the enhancement of SNOMED-CT.

To get a feeling about the structure and content of the SNOMED-CT an online browsing tool is available at (http://browser.ihtsdotools.org). For this research, I have downloaded and installed the SNOMED database and other necessary relevant tools locally. One of the tools that was used is called cTAKES. This tool is described next in detail.

5.6.2 Information Extraction by cTAKES

The Apache Clinical Text Analysis and Knowledge Extraction System (cTAKES) is a natural language processing (NLP) system for extraction of information from clinical narratives stored in electronic medical records. The cTAKES is a modular system, which has pipelines to process free text, and it uses Unstructured Information Management Architecture (UIMA) as an architectural and software framework. cTAKES was created and tested on the clinical notes from the Mayo Clinic EMR (Savova et al., 2010).

The cTAKES NER component has a terminology agnostic dictionary look-up algorithm that extracts noun-phrases from a lookup window and the extracted entity is mapped to a concept from the terminology (Savova et al., 2010). The terminology is usually the UMLS. For this research, I have coded a web service that takes a sentence and output snomed parsed XML data, which shows UMLS concepts, phrases etc. in XML format. The software was developed with Apache cTAKES SDK and Eclipse. The web service was consumed by PHP script. This way all the heavy lifting was done by the Java code and quick prototyping was done with PHP.

5.6.3 Parsing Text with cTAKES

The objective was to find snomed concepts in the GRiST clinical comments and explore any potential patterns. Firstly, I have parsed the sentences of each assessment with SNOMED-CT parser, a custom implementation based on cTAKES API library. It gave an XML output. A procedure was developed to extract the desired information from the XML tree. Concept phrases, their SNOMED-CT codes and UMLS codes have been

extracted from the XML data. The hierarchy of the concepts were also extracted by a separate procedure by traversing the concept relation table.

Giving a detailed explanation of the UIMA specification and JCas (Java Common Analysis System) object is outside the scope of this document. A detailed description can be found in (https://uima.apache.org/). In simple words, it is a XML format that contains parts of speech (POS) tag, UMLS code and other information. When we input a sentence, we get a XML output for that sentence. The system was coded according to samples provided in cTAKES SDK.

The annotation provided by the cTAKES tool includes many data types. A full list can be found in (http://ctakes.apache.org/apidocs/trunk/org/apache/ctakes/typesystem/type/textsem/IdentifiedAnnotation.html). From the annotation, I have considered the following types as relevant for our purpose. Most of the other types are related to date, measurement, roman numerals etc.

- SignSymptomMention,
- DiseaseDisorderMention,
- MedicationMention,
- ProcedureMention,
- AnatomicalSiteMention

The concept terms and their attributes [phrases, SNOMED-CT code, UMLS code, root category node] are extracted from the XML tree and stored in the database. All the relevant information including assessment_id, GRiST node names and suicide risk scores were also stored. This data was used later for finding patterns.

5.6.4 SNOMED-CT Concepts in GRiST

Approximately 50,000 assessments from between 2010 to 2014 have been chosen that had at least 500 bytes of text in its comment. Only bigram phrases were considered as

they convey a more specific meaning. The presence of average numbers of snomed concepts has been queried from the database.

Table 35 Average unique snomed occurrence

Root Category	Unique Phrase	No of Assessment	Average
AnatomicalSiteMention	6051	4122	1.47
DiseaseDisorderMention	86503	37531	2.30
MedicationMention	146	131	1.11
ProcedureMention	10159	7373	1.38
SignSymptomMention	114488	45121	2.54

Result in grapthical format:

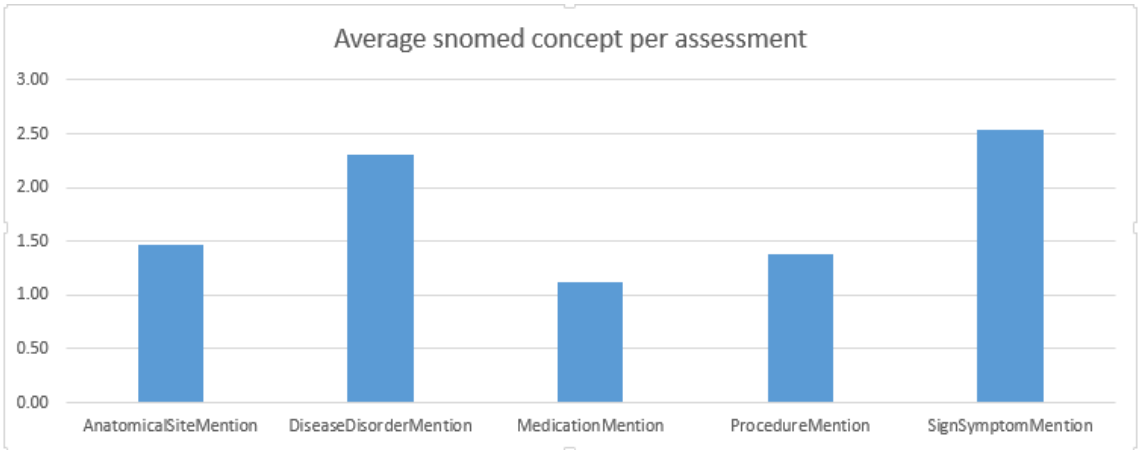

Figure 12 Average snomed concepts presence per assessment

From the above data and graph we can observe that category “SignSymptomMention” and “DiseaseDisorderMention” occurred more than the others. The result was expected as GRiST is a mental health assessment tool and the signs and symptoms expected to appear more than the medications and procedures.

Previously, phrase compression has been described with string match and vector based semantic stemming. The same method was run on snomed data and following table shows the results.

Table 36 Phrase stemming results

Snomed Type	phrase	String concept	Vector concept
AnatomicalSiteMention	490	232	138
DiseaseDisorderMention	3567	1266	676
MedicationMention	59	26	25
ProcedureMention	1234	425	289
SignSymptomMention	4485	1236	868

From the data we can see that the proposed semantic stemming can reduce the number of phrases significantly (e.g. from 4485 to 868). This again proves the utility of the proposed semantic stemming method.

5.6.5 Suicide Risk and Concept Type

Exploration work was done to find the patterns that may indicate the relationship between suicide risk and the presence of certain snomed concept types. Using a scale of 0-10 for suicide risk is too granular so I have divided the suicide risk by 3 and rounded it to get 0,1,2,3 (four) risk categories. Then the average number of times each specific snomed category occurrs per risk level was calculated. The following table shows the results.

Table 37 SNOMED category per suicide risk level

Risk	AnatomicalSite	DiseaseDisorder	Medication	Procedure	SignSymptom
0	1.41	2.02	1.09	1.31	2.13
1	1.37	2.41	1.10	1.30	2.74
2	1.43	2.68	1.00	1.33	3.02
3	1.63	2.74	1.00	2.05	2.48

For AnatomicalSite concept type, the difference is not significant across risk levels. Though a tendency of increasing with risk is present but we probably cannot rely on this for risk classification task. For DiseaseDisorder type, we can see that there is a steady growth of risk as more and more of this concept type appears in the comments. But we have to be careful here - the trend we see is actually the global average. How it works for an individual assessment is probably another matter altogether.

For MedicationMention, we can see quite similar results as we see with AnatomicalSite. Therefore, the same explanation applies to this type.

For ProcedureMention, again the number of presence increases with risk.

For SignSymptom, we can see a similar trend whereby the number of concepts increases with the risk. We can see a slight anomaly for risk level 3 whereby number of concepts is reduced slightly. The data is shown in the graph below.

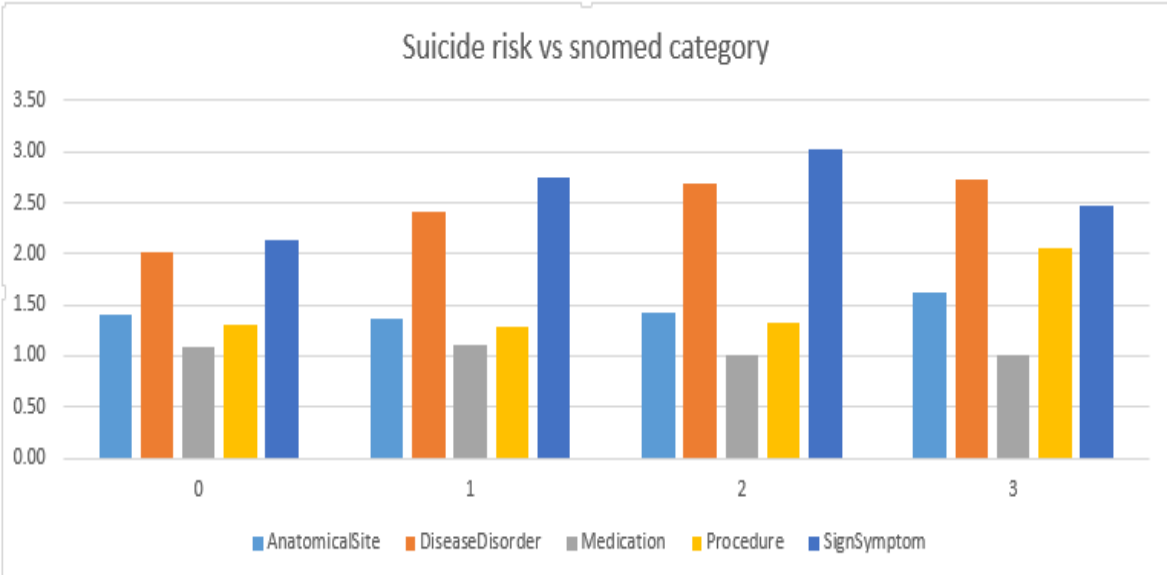

Figure 13 Suicide risk vs snomed category

Graph for all concept type is shown below.

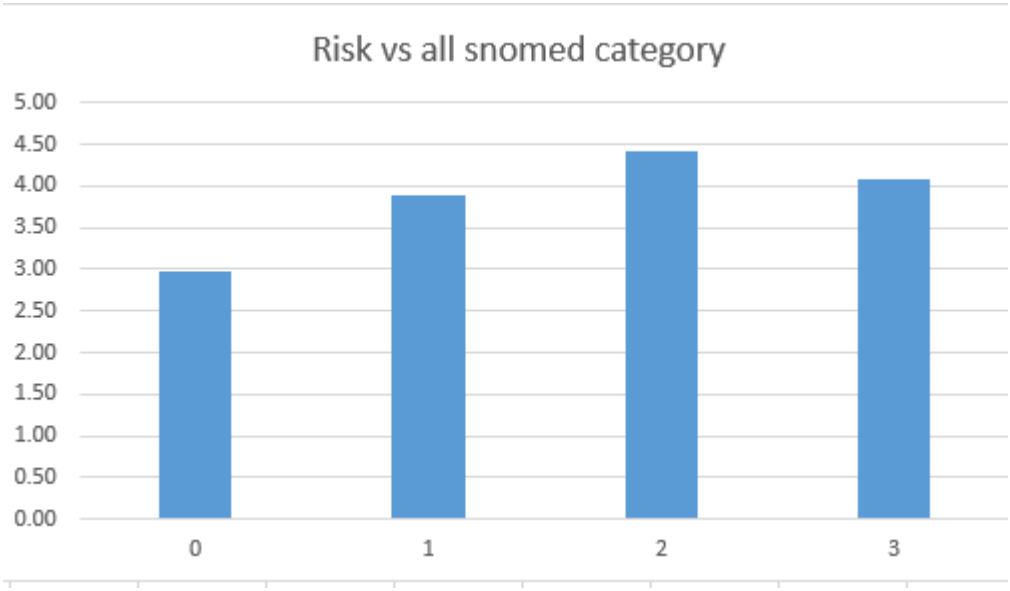

Figure 14 Risk vs snomed category

From the above graph, we see that for risk level 2 to 3 the number of concepts does not increase. In other words, if the suicide risk rating is 6 or more then there seems to be no increase in the number of mentions of the snomed concept. This analysis shows that the number of concepts might be an indicator of a higher risk category. However, fine grain classification may not be possible from the extracted concepts.

The overall conclusion is that the mentions of snomed concepts in the text tend to increase in parallel with the risk score. Higher risk patients would have more mention of various snomed concepts in their comments.

It has been hypothesised that the increased presence of snomed concepts in the comments could indicate that the patient is of a high-risk category. To validate this hypothesis a test was run on the 50,000 assessments. An assessment was queried from the database then the numbers of snomed concepts present in the comments were counted. Based on the number of concepts found a risk score is given to the assessment and compared with the clinicians given risk. High risk patient means the risk score given by the clinician is more than five and low means a score less than or equal to five.

If in an assessment, the number of concepts present was more than 2 and if we consider them as high risk then in comparison with the clinician given risk, we get a precision of 0.14 and a recall of 0.58. However, if we only compare with the patients whose clinician given risk was at least 3 then within this subgroup precision is 0.36, recall is 0.58 and F1-score is 0.45.

If the number of concepts was more than 1 and if we consider them as high risk, then in comparison with the clinician given risk precision is 0.14 and recall is 0.76. But if we only compare with the patients whose clinician given risk is at least 3 then within this subgroup precision is 0.36, recall is 0.76 and F1-score is 0.49. For clinician given scores of at least 4 we obtain precision of 0.56, recall of 0.76 and F1-score of 0.64.

These results show that this method only works when the clinicians given risk is above a certain threshold. One suitable application for this could be to alert the clinicians if many snomed concepts are found in the comments.

5.6.6 GRiST to SNOMED-CT Mapping

For further exploration purposes, we have mapped each GRiST node to the SNOMED-CT concepts. A system based on OpenMRS, an electronic medical record system framework, was enhanced by mapping its knowledge with SNOMED-CT that facilitated deploying reasoning techniques (Halland, Britz, & Gerber, 2010). To increase semantic interoperability Health Level 7 (HL7) standard has been mapped to SNOMED-CT (Ryan, 2006). A review of different techniques of mapping an ontology to SNOMED-CT can be found in (Cardillo, 2015).

BioPortal (http://bioportal.bioontology.org) is an open web accessible repository of biomedical ontologies (Whetzel et al., 2011). It provides information about the inter connection of different ontology nodes. Many ontologies in this repository are mapped to SNOMED-CT. MIMapper is a system implemented by using WordNet and mutual information between data instances to map ontologies and is found to perform better with an average F-measure of 0.84 (Kaza & Chen, 2008).

Presence of snomed node and GRiST node in comments are already available in the database from previous analysis. An SQL query was run to find the most common snomed code per GRiST node. Then the resulting snomed node was converted to its short description string. Please refer to the Table 83 in appendix B for more data.

From the data, we find that most of the nodes are semantically attached to relevant snomed nodes. It shows that this simple technique may work. A few examples are shown in the table below:

Table 38 GRiST node to SNOMED-CT node mapping

GRiST node	SNOMED-CT node
gen_coping_abil	Stress
gen_decision	Interested
gen_depression	Mood disorder of depressed type
gen_detached	Agitated (& symptom)
gen_distress	Distress
gen_jealous	Jealousy
suic	[X](Intentional self-harm) or (suicide) (event)
suic_curr_int	Thinking, function (observable entity)
suic_discovery	OD - Overdose of drug

Please note: Meaning of the GRiST node can be found in Appendix A.

We can align the GRiST ontology with the SNOMED-CT. This data tells us which node is more similar to which snomed node. This was not the primary focus of the research, which is why only a high level analysis was done for exploration purposes. Any further explorations and improvements of this technique have been left for future research. Nonetheless, this was the first time we have compared the two ontologies side by side and it can be considered a contribution to the GRiST project.

5.7 Summary

In this chapter, we have described the activities related to semantic vector representation of a phrase, a patient and a GRiST node. Many distinctive experimental works have been carried out that might help to improve the GRiST system. Most of the experiements were exploratory and aimed towards finding potential patterns in the data.

A simple technique was demonstrated in finding similar words using only Stanford dependency relationships. The result was compared with vector based similarity results. A novel method is described in this chapter, which can extract phrases from the text when no seed phrases are available. The technique can be used to build lists of phrases relevant to any specific GRiST node automatically.

To reduce the number of phrases to a representative short list, a semantic concept stemming method has been discussed in detail. A comparison has been shown between string similarity and semantic similarity with data from the GRiST system. Application of both of these techniques to reduce the number of phrases and a detailed analysis of it could be considered a contribution to NLP research.

Building the semantic profile of a patient was examined in detail. The possible application and limitations of the OpenIE system in this context is discussed using experimental data. The document vector approach is regarded as a method to build a semantic profile of a patient. It is more flexible and easy to use mathematically. The number of phrases to represent a patient can vary but if we use semantic vectors then we can use a fixed number of attributes.

I have shown how we can use word vector and build a semantic representation of the GRiST nodes. Many detailed experiments have been carried out to semantically analyse the GRiST nodes and their inter-relationships. Using word embedding to find node relationships is a new approach and I believe it is a contribution to the GRiST project. We can find the presence of a node in a text by comparing its semantic distance from the phrases in the text. This approach is generic and can be applied to other similar ontology.

Finally, the presence of SNOMED_CT concepts in GRiST data has been analysed. It has been found that the number of snomed concepts increase as the risk level increases. Mapping between GRiST and SNOMED_CT has been carried out for exploration purposes. This could help to further our understanding of the GRiST data and assist in future research.

The ultimate underlying desire of phrase extraction and semantic processing was to find patterns and help predict suicide risk. The presence of snomed concepts and other numeric attributes have been used to predict suicide risk, which is discussed in the following chapter.

6 Risk Prediction

6.1 Introduction

This chapter describes the experimental efforts to predict suicide risk from the text, the semantic vector of the assessments and from the numerical data. Many previous research papers described suicide risk predictions, disease predictions and sentiment analysis from electronic medical records, which are described in the literature review. Few of them have specifically focused on suicide risk. Some researchers worked on a limited set of data for risk prediction from textual data. Many well-known text classifiers were used to assess how well they perform in the task of suicide risk classification, especially with GRiST data. We have used raw text, SNOMED-CT concepts in the text, a semantic vector representation of patients and numerical data.

Within the GRiST system, patients are sometimes assessed multiple times. Another particular interest was to find out how the assessment varies over time by looking at the trend in the node value changes in GRiST. This might help us to measure the effectiveness of a potential clinical intervention and prompt clinicians accordingly. This could aid us to make the GRiST system more interactive. The following sections first describe risk predictions and then repeat assessment related experiments.

6.2 Methodology for Risk Prediction

Five types of experiments have been conducted to predict suicide risk from the GRiST data.

Method 1 (Using raw text): Existing toolkits like Mallet, Stanford classifier, Libshorttext and fastText were used to predict risk directly from the raw text data. Risk prediction was considered as a classification task.

Method 2 (Using phrases): The extracted phrases from Chapter 4 were used to build classifiers using the Weka machine learning tool. Weka implements many machine learning algorithms and many of them have been tried.

Method 3 (Using vector): Firstly, we created a semantic vector representation of a patient by using the document vector technique and then classified them using various algorithms.

Method 4 (Using SNOMED-CT): SNOMED-CT concepts were extracted from the clinical comments and then they were used for risk prediction.

Method 5 (Using numeric data): Numerical data has been used directly from the GRiST system, which had been inputted by the clinicians.

A detailed review has been carried out of the different well-known tools and methods. For exploration purposes, full text, extracted phrases, SNOMED-CT nodes, wordvector and various other methods have been tried with a relatively big dataset. Experimental results and their critical reviews are provided in the following sections.

6.3 **Dataset**

For this experiment, the same dataset from the GRiST database was chosen, which was used in the regression analysis. There was a total of 46903 instances of assessments of which 38197 had suicide risk of less than 5 and 8706 had suicide risk of more than or equal to 5. We considered the later group of patients as high-risk ones.

There were 21,203 assessments conducted between 2011 and 2013 that were used for training and the remaining 25,700 were used for testing. The following table shows the distribution of risk levels in the data.

Table 39 GRiST assessment data with risk level

Year/Risk	1	2	3	4	5	6	7	8	9	10	total
2011	802	1037	1148	542	513	216	214	127	49	15	4663
2012	1271	1547	1413	706	577	274	270	165	59	19	6301
2013	2824	2661	1839	995	846	381	368	228	74	23	10239
2014	4007	3228	2191	1208	928	411	395	260	86	33	12747
2015	4576	3017	1948	1237	929	425	361	297	114	49	12953
total	13480	11490	8539	4688	3793	1707	1608	1077	382	139	46903

A more detailed description of the dataset is provided in Chapter 3.

6.4 Predictions Using Full Text

This section describes the prediction of risk from raw text data. All the comments in the various nodes of an assessment were put together to build a document. These documents were then used by the following text classifiers. The classifiers were run on all the 21203 assessments conducted within 2011 to 2013. For different classification methods such as MaxEntropy, NaïveBayes and Support Vector Machine (SVM), we have used well known tools, which are described below.

6.4.1 Predictions Using Mallet

MALLET (MAchine Learning for LanguagE Toolkit) is an open source Java based package designed for statistical natural language processing, document classification, clustering, topic modelling, and information extraction. It includes tools to convert text to features, uses various machine learning algorithms (including Naïve Bayes, Maximum Entropy, and Decision Trees) for classification purposes and has a built-in classification performance evaluator (McCallum, 2002).

From all the 21203 instances, 70% of the assessments were used for training and the remaining 30% were used for testing purposes. Before running the tool, we had to

convert the data from a normal CSV format to a mallet specific format. The Mallet tool has a command line option enabling us to do that. Maxentropy and NaiveBayes were used as classification algorithms. For the first batch, a risk scale range from 1 to 10 was used and for the second batch the risk was reduced to only three categories (0=low. 1=medium, 2=high).

Table 40 Results from the Mallet Classifier

Category	Accuracy Training	Accuracy Test
Category 10 MaxEntropy	0.99	0.27
Category 10 NaiveBayes	0.60	0.30
Category 3 MaxEntropy	0.99	0.69
Category 3 NaiveBayes	0.82	0.68

We could see from the results that though training had a good accuracy score, however the test accuracy is very low (30%). When we have reduced the risk range from 10 to 3 then the test accuracy increased to 68%. Closer inspection shows that it failed to classify most of the high risk category patients. High risk category means the clinician given suicide risk is equal to or more than 5. For a screenshot of the program output, which contains a confusion matrix, please refer to Appendix C.

6.4.2 Predictions Using Stanford Classifier

The Stanford Classifier is a Java implementation of a maximum entropy classifier (otherwise known as softmax classifiers) by Manning & Klein (2003). If a training dataset with classes and textual data is provided, the classifier can automatically extract features and create a model. This model can then be used for classification of unknown data. The classifier can work with numeric real values or categorical inputs, and supports several machine learning algorithms (Manning & Klein, 2003).

The same dataset was used for this experiment, which contained 21203 assessments. Again, data was pre-processed to the format as required by this tool. For 10 classes the F1-score was 0.30 and for 3 classes the F-score was 0.47. This was again proving to be a very challenging task with this dataset. It seems that the off the shelf tools may not perform accurately enough for risk prediction. Please refer to Appendix C for the screenshot of the output.

6.4.3 Predictions Using LibshortText

LibShortText is an open source library for short-text classification and analysis. It uses LIBLINEAR classifier, which is a linear Support Vector Machine (SVM) based library. The package includes effective text pre-processing and fast training/prediction procedures (Yu, Ho, Juan, & Lin, 2013). This tool is especially suitable for short text like Twitter sentiment analysis. It comes as a set of Python programs. Data from the database was extracted by PHP scripting and then fed to the program via a shell script. This allowed us to re-run the test dynamically. Out of all the other tools, this tool seems to work faster. The following is the confusion matrix of the test results.

Table 41 Confusion matrix of LibshortText results

Risk	Original>	1	2	3	4	5	6	7	8	9	10
1	1223	672	347	144	29	17	8	6	0	0	0
2	1594	579	539	300	81	60	12	17	4	2	0
3	1746	424	595	475	114	86	10	30	12	0	0
4	833	145	252	213	107	81	9	19	7	0	0
5	740	111	189	210	86	84	19	29	12	0	0
6	332	52	54	77	52	53	11	21	11	1	0
7	304	22	55	70	41	42	18	47	8	1	0
8	198	12	35	27	23	31	15	36	17	2	0
9	72	5	8	18	5	9	7	12	7	1	0
10	26	3	1	6	1	2	1	6	6	0	0

Accuracy = 27.63% (1953/7068)

When we run the same experiment with 3 categories of risk (low, medium, and high) we can achieve a score close to 71% accuracy. The following is the confusion matrix of the 3 category tests.

Table 42 Confusion matrix of LibshortText 3 category

Risk	Original	0	1	2
0	4563	4052	506	5
1	2209	1270	930	9
2	296	88	193	15

Accuracy = 70.69% (4997/7068)

LibShortText tool's execution speed and classification accuracy was better than the other tools. This tool is very easy to run and has been run from PHP script as an external program. It can be used as a background tool to predict suicide risk in real time. For low number of risk categories, it provides a high accuracy score. Unfortunately, as the other tools described before it also failed to accurately predict most of the high risk patients. This is because the number of low risk patients was significantly higher than the number of high risk patients in the training dataset. This may have caused the classifier to become biased towards the low risk categories. Using an equal number of patients in each category did not improve the results.

6.5 **Predictions Using Extracted Concepts**

Clinical concepts were extracted from the comments and saved in the database as described in chapter 4. These extracted concepts were then used to predict suicide risk. The following section describes the experimental procedure and outlines the test results.

6.5.1 Experiments and Methods

The key phrases were extracted from the above mentioned dataset by different phrase extraction methods and saved in the database as described in Chapter 4. A Weka ARFF file was created from the extracted phrases and processed by the Weka tool. The Waikato Environment for Knowledge Analysis (WEKA) is a collection of machine learning algorithms suitable for data mining (Hall et al., 2009). This tool can be used directly as a GUI application or called from Java. It includes data pre-processing, classification, regression, clustering, association rules, and a visualisation facility (Hall et al., 2009). Weka is an open source software issued under the GNU General Public License.

Weka uses an ARFF file format, which is a text file with a header that includes the attributes name and type followed by a data portion that includes comma separated data. In our case, each line represents an instance of a suicide risk assessment. The Weka tool has a huge number of machine learning algorithms. Once we have data in the ARFF file format, we can easily apply any of these algorithms. Though I have used almost all of the algorithms to explore the data, I have only reported here the Naïve Bayes results. All the other algorithms have produced similar results. Naïve Bayes is a probability-based classification algorithm. It is a very well-known algorithm hence a description is not given in this report.

There were quite a few experiments conducted with the extracted phrases. Sometimes these experiments used the full 10 risk category classes and other times, only three (low, medium and high) classes were used. A brief introduction of each of these experiments and datasets are given below.

Experiment 1: SNOMET-CT 10 classes

This experiment uses SNOMED-CT concepts found in the comments. Risk category spans from classes 1 to 10.

Experiment 2: SNOMET-CT 3 classes

This experiment uses SNOMED-CT concepts found from the comments. The original risk was converted to three categories, low, medium and high. The risk was divided by 4 and floored to an integer value. Less than 4 is converted to 0, 4 to 7 is converted to 1, and 8 or above is converted to 2.

Experiment 3: N-gram with ECM phrasness, 3-classes

N-gram extracted phrases were filtered by the ECM phraseness method. The original risk was converted to three categories, low, medium and high.

Experiment 4: N-gram with full ECM, 3-classes

N-gram extracted phrases were filtered by the ECM phraseness and semantic filtering method. The original risk was converted to three categories, low, medium and high.

Experiment 5: ECM Phrasness string stemmed 3 classes

N-gram extracted phrases were filtered by the ECM phraseness algorithm and compressed by string stemming as described in chapter 3. The original risk was converted to three categories, low, medium and high.

Experiment 6: ECM Phraseness vector stemmed, 3 classes

N-gram extracted phrases were filtered by the ECM phraseness algorithm and compressed by vector stemming as described in chapter 3. The original risk was converted to three categories, low, medium and high.

The next section provides the experimental results. Screen shot of each of the experiments are provided in appendix C.

6.5.2 Results of the Experiments:

The following table includes a summary of all the previously mentioned phrase filtering experiments.

Table 43 Suicide risk predictions with extracted phrases

Experiment	Classes	Correct%	Precision	Recall	F1-score
E1. Snomed 10 classes	10	23.23	0.21	0.23	0.19
E2. Snomed 3 classes	3	63.01	0.58	0.63	0.59
E3. N-gram with ECM phrasness	3	57.59	0.59	0.57	0.58
E4. N-gram with full ECM (phraseness + semantic)	3	59.35	0.59	0.59	0.59
E5. ECM Phrasness string stemmed	3	61.32	0.58	0.61	0.59
E6. ECM Phraseness vector stemmed	3	60.20	0.59	0.60	0.59

From the above results, we can see that all the phrase extraction methods produce similar results. For 10 categories of risk, the accuracy was very low. For 3 categories, the accuracy was higher. However, using accuracy alone as a measure is misleading as we have seen that many high category risk assessments were not classified correctly. A screenshot of the Weka output is provided in Appendix C. From the above experimental results, we can see that the n-gram phrases filtered via the ECM method produced a similar (f-score of 0.59) result as produced by the SNOMED-CT concepts extracted by the cTAKES. This shows that a simple n-gram with filtering can be a viable alternative method to the SNOMED-CT database.

6.6 Predictions with SNOMED Code

The SNOMED-CT codes extracted from the comments in the assessments were directly used for this experiment. Their presence was used to determine the patient's risk level. With NaiveBayes classifier (10-fold validation using Weka), we have found accuracy of 63.47% and an F-score of 0.58. The test results were not different from the other methods that we have mentioned above. For this experiment, we have created a web service that can provide snomed XML format data from a sentence. This tool can be useful to other researchers.

To address the class imbalance problem, I have used the Weka tool to resample the dataset. A resample bias value of 0 leaves the class distribution as-is, a value of 1 ensures the class distribution is uniform in the output data. For a bias value of 1 or an equal number of classes in each risk category, the accuracy was 38%. For bias 0.5 the accuracy was 48%.

6.7 Predictions with Document Vector

A fixed length features vector is required by many machine learning algorithms (Le & Mikolov, 2014). Though a fixed length features vector can be created with bag-of-words, however they suffer some weakness. The bag-of-words technique used in natural language processing loses the order of words and they also ignore the semantic of the words (Le & Mikolov, 2014).

A document vector can be created by combining word vectors of the document. An algorithm is proposed by Le & Mikolov (2014) to create a document vector, which may potentially overcome the weaknesses of the bag-of-words models. "Empirical results show that Paragraph Vectors outperform bag-of-words models as well as other techniques for text representations. Finally, we achieve new state-of-the-art results on several text classification and sentiment analysis tasks" (Le & Mikolov, 2014, p. 1).

For this experiment, I have created a text file that includes suicide risk as a class and aggregated all comments as text data. Each line of the text file represents a single assessment. Then the file was processed by the python program as available from GitHub (https://github.com/klb3713/sentence2vec). This program is an implementation of the algorithm proposed by Le & Mikolov (2014). This method involves retraining of the neural network with paragraph vectors as an additional input.

After running the program, we can obtain the suicide risk and its corresponding vector (created from the text data). The generated file was then converted to the Weka ARFF file format and processed by the Weka software. Weka implements many machine learning algorithms and all major algorithms were run on the dataset. These are very common well-known algorithms so descriptions of each of them are not given.

The test results were not different from the other methods discussed above. One advantages of the document vector is that we can achieve similar results without phrase extraction. This can simplify the whole process. The following table shows the results (66% for training and 33% for testing using word2vector trained on PubMed dataset).

Table 44 Document vector classification results

Algorithm	Accuracy %	Precision	Recall	F1-measure
NaiveBayes	66.18	0.60	0.66	0.60
LibSVM*	68.47	0.46	0.68	0.55
Logistic*	68.74	0.62	0.68	0.58
IBk	0.59	0.59	0.60	0.60
DecisionTable	68.47	0.47	0.69	0.56
RandomForest	66.9	0.57	0.66	0.58

Recently another tool has become available from the same group of researchers called FastText. FastText allows us to create word embedding very fast (Bojanowski, Grave, Joulin, & Mikolov, 2016). Another component of the tool is text classification (Joulin, Grave, Bojanowski, & Mikolov, 2016). With this tool, we do not need to create a word or a document vector separately. I have installed and used this tool and tried to classify the suicide risk from comments. Unfortunately, this did not provide any better results.

6.8 Predictions with Node Similarity

Calculating the semantic vector of each GRiST node has been discussed in Chapter 5. One of the ideas was to calculate how close a specific assessment is to a given node and use that as an attribute to predict risk. For example: If we have a node ‘self harm’ and we first create a vector representation of this node by taking the average of the words vector within this node. We can then calculate how close an assessment to this specific node is by calculating the cosine similarity between the node and the assessment vector. The full method is described below:

Prediction with node vector similarity:

Step 1: Calculated the semantic representation of each node.
Step 2: For each node of an assessment, calculate the node vector by averaging the word vectors of the words that appear in that node in that assessment.
Step 3: Calculate cosine similarity for each corresponding GRIST node with assessment’s nodes.
Step 4: Use the node similarity measure as an attribute for machine learning.

The idea was to use node similarity rather than the mg-value to predict suicide risk. The simple linear regression analysis has been done with the generated data using the Weka tool. The output showed a correlation coefficient of 0.296 with the original clinician given risk value. The screenshot of the results are given in Appendix C.

The accuracy of this exploratory method is not any better than the other methods previously described in this chapter. Because vector generation depends on the text data, any missing text data would have a negative impact on the accuracy. The problem of missing comments within GRiST is discussed earlier. Although performance was not ideal, I still believe the method itself has its merits. It can help to move text analysis from string matching to numerical analysis quickly and having numerical data is desirable for analysis purposes as noted by Mikolov, Yih, et al. (2013).

6.9 Prediction with Numerical Data

In the GRIST system, the risk data is collected and stored in a database table. This collected data includes node values given by clinicians as well as textual comments. The previous sections described the risk prediction by using text data, and this section discusses the prediction of suicide risk based on the numerical data in the GRIST assessments.

Previously there have been a few other attempts made by the other members of the GRiST research team to make a risk prediction from the numerical data, but the accuracy achieved was poor. This research requires a consensus risk and without manual re-evaluation, a calculated risk was thought to be a good alternative. For this reason, at the early stage of this research I have attempted to calculate suicide risk from the numerical data.

The main challenge was handling the missing data. As GRiST has plenty of assessments already in its database, I have chosen to subgroup the assessments based on the data present and carry out the calculation only on the subgroup. This achieved a good result. The following section describes the dynamic risk prediction methods in detail with rationale.

6.9.1 Dynamic Regression

GRIST CDSS does not force clinicians to input all the data, which results in having lots of missing data in a completed risk assessment. GRiST has the following data types in its nodes.

a) Yes/NO data
b) Categorical data
c) Numeric data
d) Date and time

Categorical data is converted to numerical data based on expert weighting. The calculated value is called MG (membership grade) values. Using the mg-value to predict suicide risk is a challenge as many of the nodes values were missing. An example node is shown below:

<node label="frequency of self-harming thoughts" code="sh-freq-ideatn" question="How often do the self-injury thoughts or fantasies occur?" values="nominal" value-mg="((DAILY 1) (WEEKLY 0.5) (MONTHLY 0.2) (LESS-THAN-MONTHLY 0))"/>

To overcome the problem of missing data a simple dynamic node selection technique was adopted.

The following are the steps used:

Step 1: Determine the nodes that have been answered.
Step 2: Query the previous data and find only instances that have the same nodes answered.
Step 3: Create a dataset with this new subset.
Step 4: If there is not enough data then eliminate a node as per a preference list and go back to step 3
Step 5: If there is enough data then do regression analysis

The node preference list can be built manually or by using attribute selection algorithms like Principle Component Analysis (PCA) or based on information gain. The whole algorithm can run in real-time and work very fast. I have used Java JSP technology to build a web interface for this. The following is the screenshot of the web interface output.

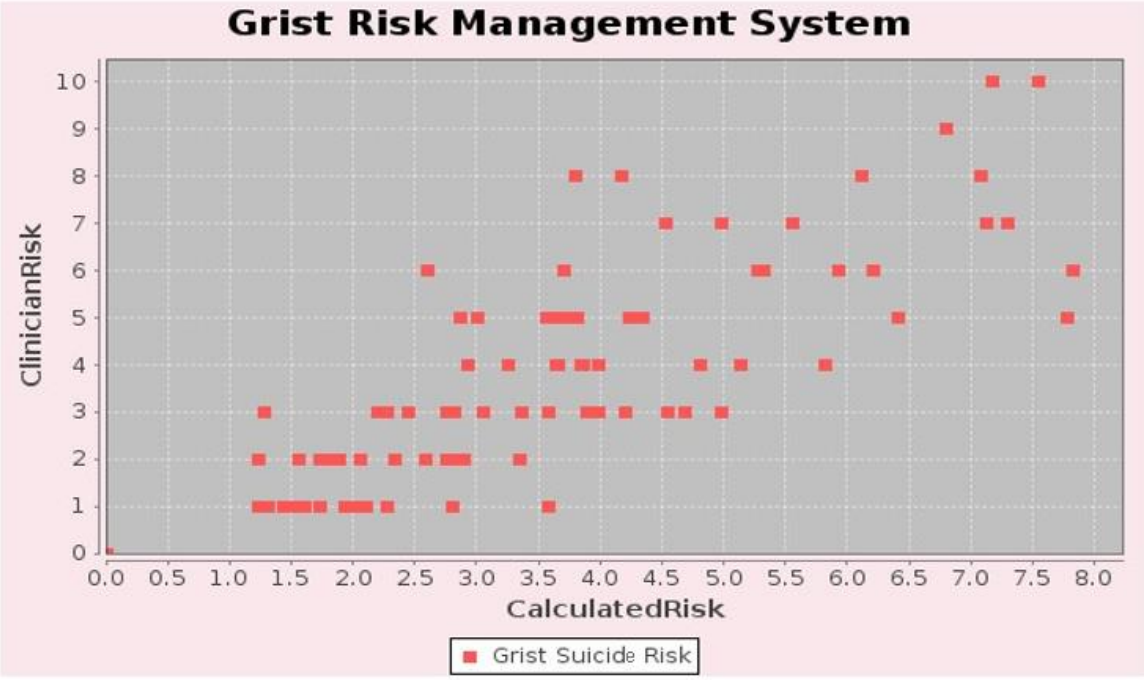

Figure 15 GRiST risk assessment by dynamic regression

The clinician given and calculated risk correlation was up to 0.92. This result was much better than anybody else had achieved on the GRiST dataset before. However, this method still failed to predict some of the patients, where too little data was available. I required a result that includes all the assessments to use it as a consensus risk for further work. To overcome this problem the following section describes an alternative approach.

6.9.2 Prediction with Scale Data Type

One of the main objectives of this research was to develop a mechanism to validate the clinician given risk data and where there was a difference between the clinician and calculated risk then explain that difference. For this, the calculated prediction value was needed for all the assessments. GRiST has a huge number of nodes (446) and they have a variety of data types as discussed earlier. For simplicity we have taken only the scale data type for which there is 141 nodes available. Then we ran the regression analysis based on these certain nodes.

The full dataset as described before was taken from the GRiST system where the clinicians given suicide risk ranged from 1 to 10. The NULL data was converted to "?" mark as per Weka ARFF format. The created ARFF file was then loaded to Weka and run with various algorithms. The linear regression analysis gave a result of approximately 0.78 correlation across all assessments by a 10-fold validation. The first 21203 assessments were used for training, the rest of the 25700 assessments were used for testing, and this gave a correlation coefficient result of 0.77. The NaiveBayes with a 10-fold validation predicted suicide risk with 40% accuracy for 10 classes and for 3 classes the accuracy was 66.62%. Resampling the data to reduce class imbalance did not yield better results.

It would be useful to find out if there were any underlying patterns that could help us to identify the cases where clinicians and regression risk differs. No one has attempted to look at this issue before with the GRiST data. We have a good amount of data, which has both clinicians and calculated risk, we can find differences between them and look for any underlying patterns. Risk difference analysis is described in detail in chapter 8.

6.10 **Analysis of Prediction Results**

The poor performance of the risk prediction could be due to three main reasons. Firstly, text can be vague and predicting different levels of risk using text is challenging. Secondly, the problem of class imbalance is due to most of the training classes being of a low risk level. Thirdly, the GRiST data has lots of missing comments.

From the experimental results, we can see that using text data and readily available classification tools to predict suicide risk was challenging. When we reduce the range to 3 (low, medium and high risk) we can see accuracy increased to 65 - 70%. But the problem is it fails to classify most of the high risk patients. In fact this result is similar to what Poulin et al. (2014) found in their experiments. They reported that they could classify patients who had suicide risk or patients who did not have a suicide risk but could not do so for low and high-risk category patients. They have used only 137 records and I have used close to 50,000 records.

This negative result agrees with other previous works. For suicidality predictions from text, predictability decreased as the data size increased (O'Dea et al., 2015). What this means is we might achieve better results where we have a hand annotated phrase list but applying that to a big dataset is still challenging.

SNOMED-CT concepts extracted by cTAKES and n-gram phrase filtered by ECM produced similar results. In fact, ECM has produced a slightly better F1-score and produced slightly better results for high risk category patients. However, the overall results are still the same and support Poulin et al. (2014) and O'Dea et al.(2015).

In the case of word and document vectors, the results were broadly similar to other text based classification. As vector only contains numerical attributes, hence they are easy to use by a variety of machine learning algorithms. For this reason, it may be preferred over text based approaches.

Class imbalance occurs in many real-world classification tasks. In class imbalanced classification, the training set for one class (majority) far surpassed the training set of the other class (minority), in which, the minority class is often the more interesting class (Ali, Shamsuddin, & Ralescu, 2015). There are different methods available for classification of imbalanced datasets, which can be divided into three main categories, the algorithmic approach, data pre-processing approach and feature selection approach (Longadge, Dongre, & Malik, 2013). We have used pre-processing to balance the risk classes, but it did not improve the outcomes.

The GRiST data contains the judgement of an individual clinician regarding a patient made on the assessment date. The risk judgement is purely an individual's judgement on a patient. Looking at the repeat assessment of the same patient shows that sometimes risk can vary significantly even within a short time span.

Another problem was that the GRiST system mainly collects numerical data and inputting comments is optional. Hence, in many instances, clinicians have given numerical input but they have not provided any comments. This resulted in lots of missing comments in the data. This is probably why numerical analysis yielded better results (up to a 0.92 correlation with the clinician given risk) than the text analysis.

6.11 **Analysis of Repeat Assessments**

This is an exploratory work to identify patterns in repeat assessments. In medical assessments, some data are permanent, and some are variable. For example, gender is a permanent attribute of a person, but weight can vary. GRiST ontology has built in concepts of variability. In this regard, two types of nodes are defined namely 'hard' and 'soft'. Some examples of hard and soft nodes are shown in the table below.

Table 45 GRiST hard and soft node examples

Hard nodes	Soft nodes
Suic_past_att (past attempt) Suic_fam_hist (famility history) hto_weapons_hist (history of using weapons) hto_any_violent (violent behaviour) sh_first_time_ep (self harm first time)	suic_regret (suicide regret) suic_how_many (suicide how many times) sh_freq_eps (self harm frequency) hto_violent (violent to other) hto_number (harm to other episodes)

Soft nodes are expected to change as time passes. This information is hard coded in the GRiST ontology based on the experts' opinion. Are all the so called soft nodes equally soft? Answering this question is important if we want to add interactivity to the GRiST system. Here interactivity means the ability of the GRiST system to suggest risk management at a granular level. For example, the system may advise clinicians to first intervene on a softer node.

Another researcher Rezaei-yazdi (2015) of the GRiST team has looked at dynamically selecting the most appropriate nodes for risk assessment. This research is different. I am trying to find the most appropriate node to intervene to manage the risk after the assessment has been completed.

When a repeat assessment is done, the hard nodes are not expected to change. The extracted data shows that not all nodes in the repeat assessment change similarly. Some nodes change more than the others do and the effects of these changes are also

different. The exploration of node changes in the context of repeat assessment may help us to design a better risk management strategy. Again, we limit ourselves to scale data type to reduce the complexity of the analysis. The experiment was run on 500 patients for whom at least 5 suicide risk assessments had been carried out. The following table shows a sample of the experimental results and for the full data please refer to Table 84 in Appendix C.

Table 46 Repeat node example data

Node Name	Suic Incre.	Suic Decr.	Suic same	Sub total	Has value	Corr. with suic	Change prob.	Probab. of decrease
gen_sad_answer	214	282	279	775	2027	0.52	0.38	0.14
suic_regret_answer	107	114	93	314	2329	0.33	0.13	0.05

From the above two example nodes we see that 'gen_sad_answer' has a total of 2027 occurrences in the sample data, and it changed 775 times in total and when it changed, suicide risk increased 214 times, 282 times suicide risk decreased, and 279 times suicide risk remained the same. The probability of change of this node is 775/2027=0.38. Probability of the risk decreasing within the changes is 282/775=0.36. If we now multiply the probability of change and the probability of the decrease, then we obtain 0.38 x 0.36 =0.14.

This value of 0.14 is the probability that this node (gen_sad_answer) will change and as a result will decrease the suicide risk. We can hypothesise that we should intervene or manage those risks attributes, which are more likely to reduce risk and have a higher probability of change. This could be incorporated in the system design to prompt clinicians for better management of the risks.

One can argue that only considering node change probability is enough for this purpose. We could also use node's general correlation with the suicide risk. A node may change more but the changes may not affect the risk much. Moreover, high correlation does not mean that the node would change, 'suic_lethality' has a high correlation with suicide risk but it hardly changes.

We investigated how the value of the node changes when risk of the present assessment is increased from previous assessments. The results are shown in the table below. I have found that the changes are not happening in the same order in conjunction with the node values and suicide risk correlation. For example, 'hto_answer' has a 0.17 correlation with suicide risk globally. But its count is increasing more times in repeat assessments, whilst the risk was increasing. In other words, 'harm to other' is likely to be found more in repeat assessments if a patient's suicide risk is increasing. In repeat assessment nodes 'sh' (self-harm), 'gen_sad' (general sadness), 'sn' (self-neglect) etc. are more likely to increase in value as the suicide risk increases.

Table 47 Risk increase and node value change

Node Name	Increase	Decrease	Total	Correlation
sh_answer (self-harm)	119	20	178	0.69
gen_sad_answer (general sadness)	101	25	222	0.52
sn_answer (self neglect)	94	25	204	0.26
vuln_su_answer (vulnerability to service user)	93	25	247	0.25
gen_anx_emotns_answer (anxious or fearful)	90	27	209	0.25
gen_helpless_answer	88	25	191	0.54
gen_life_not_livng_answer	88	19	165	0.63
gen_distress_answer	88	31	210	0.43
gen_negative_self_answer	85	21	182	0.49
hto_answer (harm to other)	80	25	174	0.17
gen_plans_future_answer (future plan)	79	21	171	0.49
gen_mood_swings_answer	78	25	200	0.42
suic_pot_trig_answer (potential trigger)	73	11	101	0.46
gen_angry_emotns_answer	72	24	177	0.25
suic_id_hi_risk_answer (suicide ideations)	62	4	85	0.86
suic_id_control_answer (suicide ideation control)	59	3	79	0.65
gen_listless_answer (loss of drives)	59	10	109	0.22
suic_id_strngth_answer (ideation strength)	59	3	78	0.78
gen_motivation_answer	58	13	107	0.31
gen_mental_withd_answer (mental withdrawal)	58	14	110	0.23
worthlessness_answer	55	9	92	0.47

Node Name	Increase	Decrease	Total	Correlation
gen_phys_withd_answer (physical withdrwal)	53	9	104	0.20
risk_dep_answer (risk to dependents)	48	16	110	0.39
gen_sleep_dist_answer	46	22	144	0.28

Please note: The meaning of the GRiST node can be found in Appendix A.

This repeat assessment analysis can help us to build a more intelligent and interactive CDSS system. The system can suggest to a clinician where to intervene to reduce risk. It can give intelligent suggestions in real-time to manage risk better. Based on the data a list of nodes can be created that would provide better results upon intervention.

For example if we have two areas where the clinician can intervene, then the system can calculate which one would probably yield better results in terms of risk reduction. This technique would allow the GRiST system to provide risk management suggestions in real time. This is a significant contribution towards the improvement of the GRiST system.

6.12 **Summary**

It could be helpful if we can accurately predict suicide risk from the clinical notes. A linguistics-driven prediction model is described by Poulin et al. (2014) to estimate the risk of suicide from clinical notes. Automatic detection of suicidality in Twitter investigated by O'Dea et al. (2015). Their sample size and scope were limited whereas we have used a bigger sample from the GRiST system. Application of the existing tools and algorithms show that more work is needed in this regard. Most of the research used human created lists of positive or negative words to conduct sentiment analysis or a list of symptoms to predict disease from text. But having generic systems to do this prediction from text data in an unsupervised manner is quite challenging.

Furthermore, we have also tried to predict risk by using the presence of SNOMED-CT concepts in the text as well as using semantic vector representation of the patients. If we

reduce the category of risk from 10 to 3 (low, medium, high) the prediction accuracy increases. However, a careful observation shows that most of the high risk patients are not predicted correctly. Our results actually support the conclusion made by other researchers such as Thompson et al. (2014), O'Dea et al.(2015) and Poulin et al. (2014) that the prediction of different degrees of risk from text is challenging. The comprehensive list of experiments carried out in this regard and their critical analysis could be useful for future research.

I have been able to predict suicide risk of about 50,000 assessments with correlation coefficient values of 0.78 using only the scale type nodes. It shows that using non-categorical or ordinal data provides better results than using artificial values for categorical data. GRiST uses mg-values, which are estimated for categorical data by experts. This experiment shows that not using them in regression analysis produces better results. The regression analysis data from this chapter is used later in Chapter 8 as a consensus risk.

Firstly, we have attempted to predict risk from text and numerical data. From empirical results, we have seen that predicting higher risk patients is challenging even though the overall accuracy might be good. Considering the challenges of predicting higher risk, we have tried to explore the GRiST node inter-relationships as the next step to predict risk. We have applied frequent itemset mining to identify high risk category patients more accurately. The next chapter discusses association rule mining in the context of GRiST.

7 Association Rule Mining

7.1 Introduction

In the literature, frequent itemset mining has been shown to be successful in detecting disease and symptom relationships. Risk prediction results discussed in the previous chapters shows that accurately detecting higher levels of suicide risk is challenging. Detecting a specific category of risk from patient data can be unreliable due to class imbalance problems even if the overall accuracy may be higher. To improve diagnostic accuracy the association rule mining technique was applied. Association rule mining is one of the fundamental research topics in data mining, which identifies interesting relationships between itemsets (S. Zhang & Wu, 2011).

Application of rule mining is challenging in risk analysis, as risk generally is a rare event. Using low support or other methods have been described in the literatures to extract rules. We have proposed a multi-rule based approach to predict risk. Application of our method demonstrates that we can predict high suicide risk with more confidence than the normal association rule mining techniques. The multi-rule approach proposed in this chapter improves prediction accuracy and is easily configurable.

It has been hypothesised that the GRiST ontology node relationships analysis might help to identify exceptional cases, especially the identification of high risk category patients. To find node relationships, various statistical methods have been explored. Then the impact of the discovered relationships on suicide risk has been analysed. We have used two distinctive approaches for node relationships identification:

1. Chi-square analysis
2. Frequent itemset mining

This chapter describes the theoretical background, the proposed new method and the experimental results.

7.2 Dataset

For this experiment, the same dataset from the GRiST database was chosen, which was used in the regression analysis. There was a total of 46903 instances of assessments in which 38197 had suicide risk of less than 5 and 8706 had suicide risk of more than or equal to 5. We considered the later group of patients as high-risk patients.

Assessments conducted between 2011 and 2013, a total of 21203 assessments were used for training and the rest of the 25700 were used for testing purposes. The following table shows the distribution of risk levels in the data.

Table 48 GRiST assessment data with risk level

Year/Risk	1	2	3	4	5	6	7	8	9	10	total
2011	802	1037	1148	542	513	216	214	127	49	15	4663
2012	1271	1547	1413	706	577	274	270	165	59	19	6301
2013	2824	2661	1839	995	846	381	368	228	74	23	10239
2014	4007	3228	2191	1208	928	411	395	260	86	33	12747
2015	4576	3017	1948	1237	929	425	361	297	114	49	12953
total	13480	11490	8539	4688	3793	1707	1608	1077	382	139	46903

A more detailed description of the dataset is provided in Chapter 3.

The following table shows the correlation coefficient between calculated and predicted risk across varying risk levels. Correlation co-efficient for the high risk category patients was much lower. It indicates that the clinicians given risk differs significantly from the calculated risk for high risk category patients.

Table 49 Correlation between clinical and calculated risk

Risk Level	Correlation
>0	0.785
>1	0.726
>2	0.661
>3	0.590
>4	0.520

>5	0.421
>6	0.341
>7	0.252

The overall correlation between the clinicians given and calculated risk is 0.78, but the predictive accuracy varies as the levels of risk increase. For the test dataset high risk (>=5) was predicted with 66% accuracy. Our objective was to find methods to identify a high-risk patient more accurately. In order to achieve this, we have used the node relationships and frequent itemset mining techniques as described in the following sections.

7.3 Node Relationships by Chi-square

Chi-square statistics was used to determine whether there was a significant association between any two GRiST nodes. The following two different sets of experiments have been carried out.

1. Nodes share similar phrases
2. Nodes share similar level of numeric values

Before providing an introduction to the chi-square method, the definition of some of the terms used in this report is provided below.

7.3.1 Explanation of the Terminology

In this chapter, some concepts or terms are repeated many times. The following are the explanation of those terms.

If we consider two hypothetical nodes nodeA and nodeB in the GRiST ontology then:

nodeA is present = there is a phrase match or numerical value match with nodeA within a specific assessment.

nodeA average risk= average suicide risk of all the assessments where nodeA is present.

Combined risk= the average of suicide risk when both nodeA and nodeB are present.

Combined risk high= combined risk is higher than any individual node's average risk.
High risk relationship= relationships whose combined risk is higher.

High risk category assessment= assessments where suicide risk is higher than a specific value (e.g. >=5).

Node relationship/pattern = when chi-square analysis indicates a statistically significant relationship between two nodes.

Pattern or relationship exists= when an assessment has a specific node relationship present in it.

7.3.2 Introduction to Pearson's Chi-square Test

Chi-square (X^2) test is a nonparametric statistical analysing method often used in experimental work, where the data consist in frequencies or counts for example, the number of people exposed and the number of them who had diseases (Zibran, 2015). The chi-square statistic can be used to test the hypothesis of no association between two attributes, groups or events. However, the statistical association confirmed by the chi-square method does not automatically imply any causal relationship between the groups being compared, but it means the relationship is worth investigating (Zibran, 2015).

In a simple case, we can use 2x2 contingency table to calculate chi-square (Zibran, 2015). For example:

Null hypothesis: Exposer and disease is independent.

Table 50 2 by 2 contingency table

	Disease		
Exposer	Yes	No	
Yes	a	b	a+b
No	c	d	c+d
	a+c	b+d	a+b+b+d

Then we can write:

$$chisquare\ (X^2) = \frac{(ad - bc)^2(a + b + c + d)}{(a + b)(c + d)(d + c)(a + c)} \quad (12)$$

To assess the significance of the calculated value of X^2, we refer to the standard chi-square table. This table contains the critical X^2 values on different degrees of freedom and levels of probability (Zibran, 2015). The degree of freedom for a 2 × 2 contingency table is (2-1) (2-1) =1.

First, we calculate the chi-square and the degree of freedom, and then we can consult the chi-square table and look into the row corresponding to the given degree of freedom. If the corresponding probability value is less than or equal to 5% then we reject the null hypothesis. Therefore, we can conclude that the exposer and disease has an association (Zibran, 2015).

The next sections describe different experiments performed using the chi-square test. If the p-value was less than 0.05 then I rejected the null hypothesis and considered there to be a relationship between the two nodes.

7.3.3 Relationship by Phrase

The SNOMED-CT concept phrases from each of the GRiST nodes were extracted and saved in the database. For each assessment, a list of nodes was created only if the frequent phrases in those nodes matched with the phrases in that assessment. For example, one assessment may have phrases that match to a particular set of GRiST nodes. If an assessment has phrases that match with phrases of a node then we assume that the assessment has semantic similarity with that particular node.

A chi-square model was created as described below

Table 51: 2 by 2 contingency table

	Node2	Not Node2
Node1	a	b
Not Node1	c	d

Where an assessment:

a= match with both node1 and node2

b=match with node1 but not node2

c=do not match with node1 but match with node2

d=do not match with node1 and node2

$$chisquare = \frac{(ad - bc)^2(a + b + c + d)}{(a + b)(c + d)(d + c)(a + c)} \quad (13)$$

Please refer to the **Table 80** in Appendix B for more data. The following table shows some of the found relationships. The meaning of the node name can be found in Appendix A.

Table 52 Node to node relationship by snomed concept phrases

Node Name	Other node	Both present	Avg Risk	Chi	p-value	remark
gen_alc_misuse (alcohol misuse)	suic_lethality	16	4.31	9.12	0.003	increase
gen_app_diet (appetite)	suic_pot_trig (potential trigger)	50	4.26	9.03	0.003	increase
gen_liv_skills (live skill)	Sn (self-neglect)	23	2.48	40.41	0.000	decrease
gen_sleep_dist	Hto (harm to other)	113	2.51	11.69	0.001	decrease

From the experimental data it has been found that there was a total of 728 relationships with p-value<0.05 which means that the relationships were statistically significant. I have calculated the average risk of each individual node and the combined average risk of the nodes (when both nodes were present). I have found that in 271 cases, combined risk is more than the individual node risk and in 171 cases, combined risk is low. Which means when there is a relationship between two nodes then the combined (when they are both present in the assessment) risk is more likely to be higher.

For each risk category, I have looked at how many times any relationship appears in the assessment and how many times high risk relationships appear. The following table shows the results.

Table 53 Node to node relationship and risk

Risk level	No of Assessment	Any relationship per assessment	High rel. per assessment	Total rel. count	High rel. count	High percent
1	4897	1.76	0.13	8632	614	7.11
2	5245	1.92	0.23	10064	1224	12.16
3	4400	1.99	0.36	8737	1593	18.23
4	2243	2.45	0.57	5503	1282	23.3
5	1936	2.53	0.62	4889	1205	24.65
6	871	2.83	0.79	2464	691	28.04
7	852	2.75	0.83	2344	704	30.03
8	520	2.74	0.87	1424	453	31.81

9	182	2.31	0.81	421	148	35.15
10	57	4.82	2.12	275	121	44

Interestingly, it has been found that as the risk increases so does the presence of high risk category relationships (where the combined node risk is high). In the risk category level_1 only 7% of the relationships were from high risk but in risk category level_8 almost 31% of the relationships came from high risk relationships. These results indicate that finding high risk relationships within an assessment is useful and we can flag their presence to the system user.

For example if in an assessment we find that there are some high risk relationships present then we can alert the clinician that the potential suicide risk for this patient is high. I have continued to investigate these relationships further with different criteria, which are described in the following sections.

7.3.4 Relationships by Node Value

Relationships found by phrase matching may not be as precise as those that are found by numerical value matching. To verify the results found by phrase matching, I have used numerical data to repeat the experiment. Here numerical data means the membership grade (mg value) that was inputted by the clinicians. In the scale datatype nodes, the values that was given by the clinicians ranges from 0 to 10.

Chi-square was calculated by using the values as defined below:

a= in an assessment both node1 and node2 has value >=5 (high value)
b= node1 high value but not node2
c= node1 low value but node2 high value
d= both node1 and node2 low value

$$chisquare = \frac{(ad - bc)^2(a + b + c + d)}{(a + b)(c + d)(d + c)(a + c)} \quad (14)$$

Please refer to the Appendix B and Table 81 for more data. The following table shows some of the results.

Table 54 Node relationships by mg-value

Node	Other node	Both count	Avg risk	Chi square	p-value	Remark
suic_discovery (suicide discovery)	suic_pot_trig (potential trigger)	1157	5.57	33.32	0	increase
hto_answer (harm to other)	vuln_su_answer (feeling vulnerable)	1713	3.14	507.49	0	decrease
suic_lethality	suic_pot_trig (potential trigger)	2763	5.29	61.93	0	increase
gen_helpless (general helpless)	suic_id_hi_risk (ideation high risk)	1278	6.35	448.03	0	increase
gen_life_not_livng	suic_id_control (ideation control)	1026	6.49	621.96	0	increase

From the data, we have found that there are a total of 595 relationships of which 443 relationships have high combined risk. Again, in this case a relationship of which combined risk is high appears significantly higher in high risk category assessments.

Table 55 Node relation and risk category by mg-value

Risk level	No of Assessment	Any rel. per assessment	High rel. per assessment	Total rel. count	High rel. count	High percent
1	4897	22.36	10.14	109503	49653	45.34
2	5245	28.24	15.78	148102	82775	55.89
3	4400	37.84	24.29	166504	106860	64.18
4	2243	55.7	38.4	124932	86133	68.94
5	1936	76.75	55.95	148591	108322	72.9
6	871	99.47	75.23	86639	65523	75.63
7	852	120.92	93.68	103026	79814	77.47
8	520	133.6	104.78	69472	54483	78.42
9	182	147.86	116.15	26911	21140	78.56
10	57	146.63	113.53	8358	6471	77.42

From the above table we can see that in low suicide risk (e.g. 1) the percentage of high risk relationships is about 45% but for high risk patients the high-risk relationship is present about 78% of the time. We can view the results in the following graph.

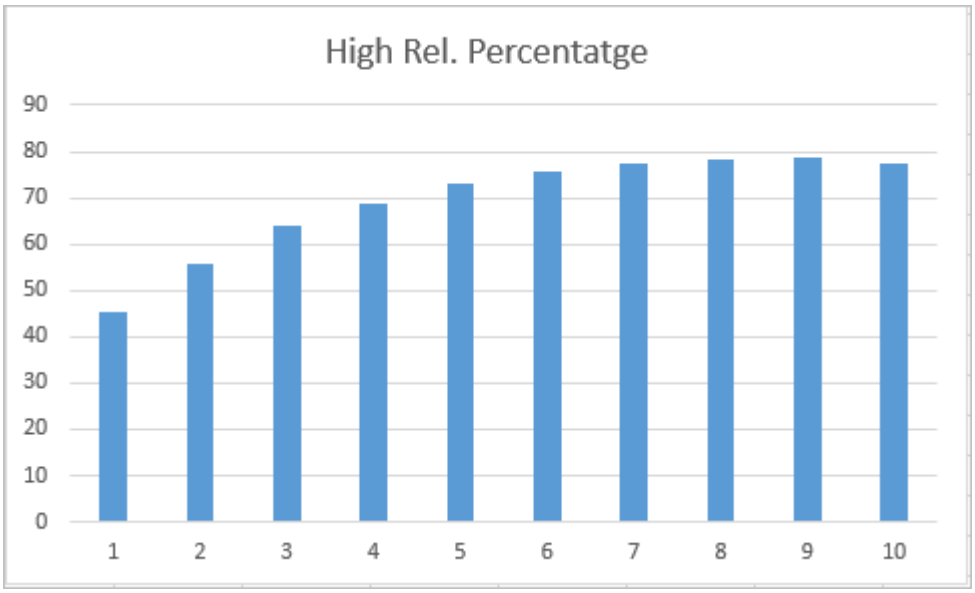

Figure 16 Risk level vs high risk relationships

7.3.5 Ch-Square Relationships Analysis

After running different sets of relationship tests and analysing the data, I have found that there is a general tendency of combined risk (when both nodes have a value in an assessment) being higher when there was a chi-square relationship between two nodes. High risk relationship types are also more likely to be present in the high risk assessments.

This is an interesting finding in the context of the GRiST system. We now know that there exists relationships between nodes and those relationships tend to produce higher risk. If we analyse a patient and find the 'high risk type relationship' then that could potentially indicate suicide risk for that patient is higher. This finding is a useful contribution towards making the GRiST system more interactive.

Considering the potential significance of the relationship between GRiST nodes, we have tried a more systematic approach to predict suicide risk by frequent itemset mining, where one of the items is always suicide risk. To do this analysis we have used an fp-growth algorithm as it was claimed to be faster in the literature. The next section discusses frequent itemset mining techniques.

7.4 Frequent Itemset Mining

Frequent itemset mining was initially introduced for market basket analysis. In the literature review, we have described the increasing use of association rule mining in disease and symptom relationships analysis. In the previous section, we have shown that relationships of nodes are more likely to appear in higher risk category patients. This suggests that application of association rule mining techniques may provide a means to identify high-risk patients. The following sections describe theoretical introduction followed by experimental results and analysis.

7.4.1 Theoretical Background

Association rule mining is one of the fundamental research topics in data mining, which identifies interesting relationships between items and predicts the associative and correlative behaviours for new data (S. Zhang & Wu, 2011). Frequent itemset mining is a popular technique that was originally developed for market basket analysis and it is now commonly used in discovering regularities between nominal variables (Borgelt, 2012).

The problem of discovering all association rules can be divided into two sub problems (Agrawal & Srikant, 1994):

1. Find all sets of items (itemsets) that have minimum support (in the number of transactions they appear).

2. Use the large itemsets to generate the desired rules.

Based on the definition given by Agrawal, Imielinski, & Swami (1993), association rules mining can described as follows:

Let I = {i1, i2 . . . in} be a set of n distinct attributes called items.
Let T= {t1, t2, t3, …..tm} be a set of m transactions.

Each transaction in T has a unique transaction ID and contains a subset of the items in I.

Let X, Y be a set of items; an association rule is an implication of the form

$X \Rightarrow Y$, where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$.

X is called the antecedent or left hand side (LHS) and Y is called the consequent or right hand side (RHS). In order to select interesting rules various measures of significance and interest are used. Two commonly used measures are the support and confidence of a rule.

Support of X is the proportion of transactions in T that contain both X.

$$Supp\ (X) = \frac{\text{Number of transations conatin X}}{Total\ number\ of\ transactions}$$

The confidence value of a rule (X =>Y) with respect to a set of transactions T is the proportion of the transactions that contains X, which also contains Y.

$$Conf\ (X => Y) = \frac{\text{Supp(XUY)}}{Supp(X)}$$

The problem of discovering all association rules from a transactional database is to generate the rules that have a support and confidence greater than predefined

thresholds. Such rules are called valid (or strong) rules, and the framework is known as the support–confidence framework (L. Zhou & Yau, 2007). In depth explanation of the association rule mining techniques can be found in (S. Zhang & Wu, 2011), (Naulaerts et al., 2015), (Hipp et al., 2000) and (Borgelt, 2012).

The performance and complexity of an association rule mining system is greatly dependent upon the identification of frequent itemsets (S. Zhang & Wu, 2011). One of the well-known algorithms to perform this identification is the Apriori algorithm. For a detailed description of the Apriori algorithm please refer to Agrawal & Srikant (1994). The Apriori algorithm is quite slow and one of the new faster algorithms is the FP-Growth algorithm developed by Han, Pei, Yin, & Mao (2004). The FP-growth algorithm is used in this research hence it is described in detail in the next section.

7.4.2 FP-Growth Algorithm

The FP-Growth algorithm is proposed by Han et al. (2004). The proposed algorithm uses a novel frequent-pattern tree (FP-tree) structure, which is an extended prefix-tree structure for storing information about frequent patterns in a compressed form. FP-growth algorithm mines the complete set of frequent patterns by analysing pattern fragment growth (Han et al., 2004).

According to Han et al. (2004) the efficiency of the fp-growth method comes from the following three techniques:

1. A condensed data structure (FP-tree) avoids costly, repeated database scans,

2. It uses a pattern-fragment growth method to avoid the costly generation of a large number of candidate sets, and

3. A divide-and-conquer method is used to decompose the mining task into small sets, which dramatically reduces the search space.

The fundamental principle of the FP-Growth algorithm is explained with an example below. This example is summarised from the original paper written by Han et al. (2004).

The following table has five transactions of items that are bought by customers. Firstly, all the items are counted and an ordered list is created. Items that have support of less than 3 are ignored.

Table 56 A transactional database for FP-tree example

TransactionID	Items bought	(Ordered) frequent items
1	f, a, c, d, g, i,m,p	f, c, a,m, p
2	a, b, c, f, l,m,o	f, c, a, b,m
3	b, f, h, j,o	f, b
4	b, c, k, s,p	c, b, p
5	a, f, c, e, l, p,m,n	f, c, a,m, p

After this an FP-tree is created using the following steps as described in (Han et al., 2004).

Step1: The first transaction has items (f,c,a,m,p). A tree is created as shown in the image below.

Step2: For the second transaction (f, c, a, b,m) shares a common prefix (f, c, a) with the existing path (f, c, a,m, p), the count of each node along the prefix is incremented by 1, and one new node (b:1) is created and linked as a child of (a:2) and another new node (m:1) is created and linked as the child of (b:1).

Step3: For the third transaction, since its frequent item list (f, b) shares only the node (f) with the f -prefix subtree, f ’s count is incremented by 1, and a new node (b:1) is created and linked as a child of (f :3).

Step4: The scan of the fourth transaction leads to the construction of the second branch of the tree, (c:1), (b:1), (p:1).

Step5: For the last transaction, since its frequent item list (f, c, a,m, p) is identical to the first one, the path is shared with the count of each node along the path incremented by 1.

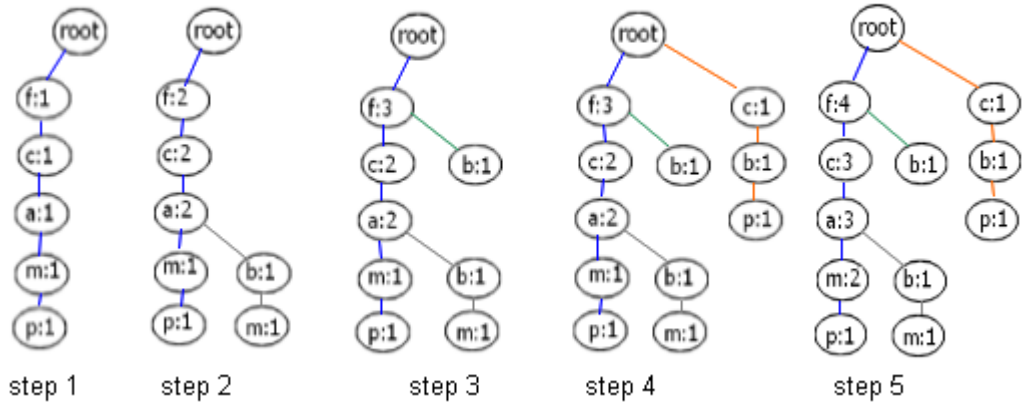

Figure 17 FP-growth creation example

Compact FP-tree creation helped to perform subsequent tasks more efficiently on a compact data structure (Han et al., 2004). All the possible patterns containing only frequent items and a node can be found by following the nodes link towards the root and starting from the node head. A detailed description of the FP-growth can be found in (Han et al., 2004). For the experimental purpose, I have used Weka tools, which has an implementation of the FP-growth algorithm.

7.4.3 Experimental Results

The Weka data mining tool was used to extract the association rules among GRiST nodes by using the FP-growth algorithm. For this analysis, 141 scale data type nodes were chosen. If the value of the node was >=5 then 1 was chosen and if the value was <5 then 0 was chosen.

The first 21203 records were used to learn association rules by using the fp-growth algorithm. The following table shows some of the learned rules with Confidence=0.85 and Support=0.05. To make it useful for our purpose I have only chosen rules that refer

to the suic_answer node. A PHP script was written to filter all the rules and find only the suic_answer related rules. The detailed meanings of nodes are given in Appendix A. Please read id=ideation, suic=suicide, pot=potential, trig=trigger, hi=high, gen=general, strngth=strength.

Table 57 Some sample rules extracted by the FP-growth algorithm

Antecedent nodes (left hand side)	Consequent node (right hand side)	Confidence of the rule
suic_id_hi_risk, suic_id_strngth, suic_pot_trig	suic (suicde risk)	0.92
suic_id_control, suic_id_hi_risk	suic	0.90
suic_id_hi_risk, suic_id_strngth	suic	0.90
gen_life_not_livng, suic_id_hi_risk	suic	0.89
suic_id_control, suic_id_strngth	suic	0.88
gen_distress, suic_id_hi_risk	suic	0.87
gen_sad, suic_id_hi_risk	suic	0.87
suic_id_hi_risk, suic_pot_trig	suic	0.87
suic_id_control, suic_pot_trig	suic	0.87
gen_helpless, suic_id_hi_risk	suic	0.87
gen_negative_self, suic_id_hi_risk	suic	0.86
gen_life_not_livng, suic_id_strngth	suic	0.86
suic_id_strngth, suic_pot_trig	suic	0.85

Please note: The meaning of the GRiST node can be found in Appendix A.

In the GRiST data, most of the nodes rarely have any values as clinician can skip nodes. The support parameter was lowered to 0.05 and the confidence parameter chosen was 0.85. This has generated a total of 144 rules. Even though support was 0.05 it still was using at least 1060 out of 21203 assessments to create a rule. The high suicide risk is a rare event hence threshold needs to be low to find any rule. Use of low support value for scarce data, especially in biomedical application is not uncommon. For example, in some experiments support 1.5% was used since mental disorders are relatively rare in the healthy population, especially for those who have compounded disorders (Lacković et al., 2014). For negative and positive association rules mining from

text using frequent and infrequent itemsets, support from 0.05 to 0.15 was used by Mahmood, Shahbaz, & Guergachi (2014). Rare events mining is discussed in further detain in section 7.5.

7.4.4 Risk Prediction by Association Rules

Once we have found and filtered the association rules from the training dataset then we have used them to predict suicide risk from the test dataset. The method of predictions is shown below:

Step1: Make a list of rules
Step2: Extract patient data
Step3: Match patient attributes with each of the rules.
Step4: If a match is found then it is high risk or else it is low risk
Step5: Repeat this for another patient

For example, for support=0.05 and confidence=0.85 a total of 144 rules were extracted. Out of which 13 rules had suic_answer high at the right hand side. We used these 13 rules to predict high risk based on the above mentioned steps.

Out of 25000 test instances the number of high risk patients was 4165, the number of predicted high risk patients was 2518 of which 1829 predictions were correct. For the high-risk patient this gives us a recall=0.439, a precision=0.726 and an f-score=0.54. The overall accuracy of prediction including high and low risk was about 88%.

The following table shows data with different confidence and support levels.

Table 58 Precision and Recall of High risk prediction

Support	Confidence	Rule count	High recall	High precision	F-measure
0.03	0.75	539	0.80	0.51	0.62
0.03	0.8	404	0.74	0.56	0.64
0.03	0.85	299	0.56	0.67	0.61
0.03	0.9	163	0.43	0.75	0.54
0.03	0.95	5	0.25	0.90	0.39
0.04	0.75	114	0.76	0.55	0.63
0.04	0.8	76	0.66	0.61	0.64
0.04	0.85	55	0.50	0.69	0.58
0.04	0.9	20	0.37	0.81	0.51
0.05	0.75	26	0.65	0.60	0.63
0.05	0.8	16	0.50	0.69	0.58
0.05	0.85	13	0.44	0.73	0.55
0.05	0.9	3	0.26	0.87	0.40

From the above test results, we can conclude that whilst the proposed method cannot always predict (low recall value) but when it can, the prediction can be up to 87% accurate. This method is very flexible and we can predict with different levels of confidence. Rules that have high confidence produce results that are more accurate as shown in the graph below. This is an important finding with the GRiST data. It could allow us to predict risk as soon as a pattern is found before the assessment is completely finished.

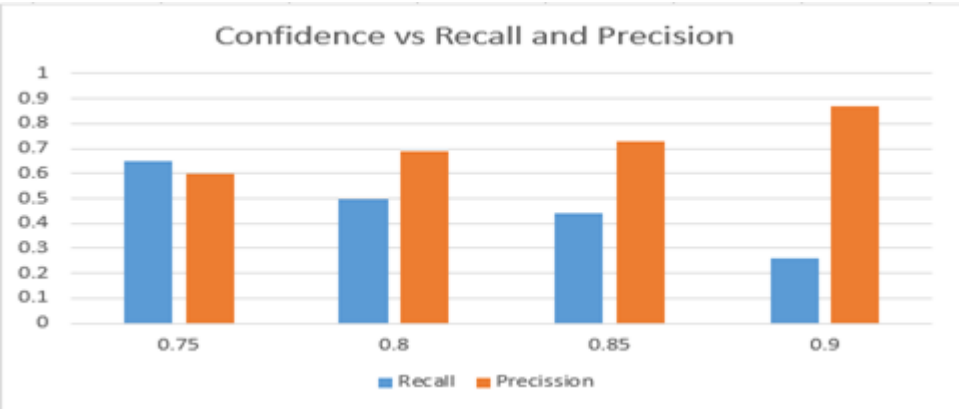

Figure 18 Association rules confidence and accuracy

One of the problems is that the support value is low due to rare occurrences of the attributes. We propose a simple solution to overcome this problem by using multiple rules for predictions. If we increase the number of required rules to predict suicide risk to more than one, then we can see that precision also increases. This new approach would allow us to modify precision and recall and achieve the desired level of accuracy. The following table shows some of the experimental data, which are filtered from rules (144) found with min support=0.05.

Table 59 High risk prediction with multiple rules

Confidence	Rule count	Min rules	High recall	High precession	F-score
0.8	16	1	0.50	0.69	0.58
0.8	16	2	0.41	0.76	0.53
0.8	16	3	0.37	0.80	0.51
0.8	16	4	0.34	0.83	0.48
0.85	13	1	0.44	0.73	0.55
0.85	13	2	0.37	0.80	0.51
0.85	13	3	0.33	0.83	0.48
0.85	13	4	0.30	0.85	0.45
0.90	3	1	0.26	0.87	0.40
0.90	3	2	0.23	0.88	0.37
0.90	3	3	0.17	0.92	0.29

The following graph shows the effect of using multiple rules extracted by using confidence 0.85.

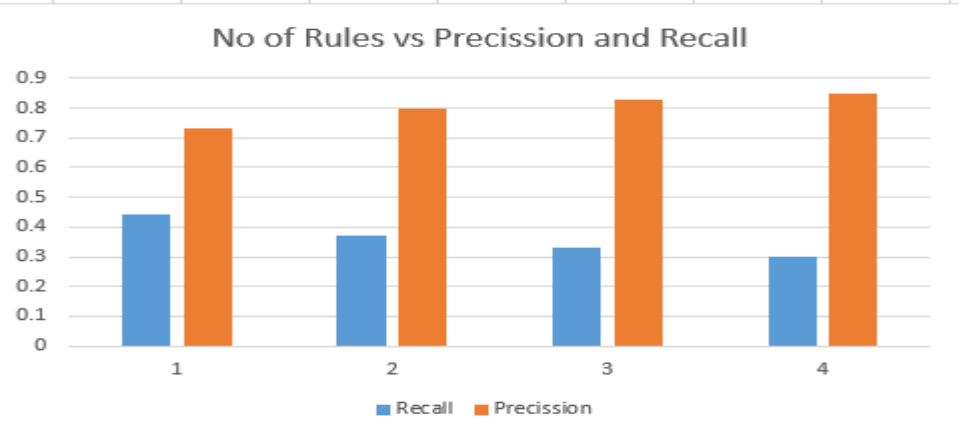

Figure 19 No of rules vs Recall and Precision

From the above table we see that while initial precision was 69% we can improve that to 83% by matching at least 4 rules. This could allow us to apply this method and alert clinicians with different confidence levels. An alert can be generated as soon as the relevant attributes become available. The system does not need to wait for the completion of the assessment. To investigate it further, we have compared our approach with other rare event mining methods in the subsequent sections.

7.5 Rare Event Mining

Rare itemset mining has a wide range of application possibilities in the field of risk assessment and fraud detection (Abraham & Joseph, 2016). It can provide useful information in different decision-making domains such as business transactions, medical, security, fraudulent transactions and retail communities (Pillai, 2010). For example in medical dataset a rare combination of syndrome plays a vital role for the physicians (Bhatt & Patel, 2015). A review of rare itemset mining can be found in (Kiruthika & Roopa, 2015).

In the GRiST dataset, higher risk of suicide is a rare event. Previously we have shown the application of fp-growth methods with low support values. The following sections describe the application of two rare itemset mining techniques on the GRiST data. As before the extracted rules were used to predict suicide risk.

7.5.1 Using the CORI Algorithm

CORI is an algorithm for discovering itemsets (group of items) that are rare and correlated in a transaction database (rare correlated itemsets). A rare itemset is an itemset such that its support is low (less than maximum support set by the user) but they are correlated amongst themselves. The support of an itemset is the number of transactions containing the itemset (Fournier-Viger et al., 2016a).

To find the rare but correlated itemset a new measure BOND is proposed by Bouasker & Ben Yahia (2015). A correlated itemset is an itemset such that its bond is no less than a minimum bond threshold set by the user. The bond of an itemset is the number of transactions containing the itemset divided by the number of transactions containing any of its items. The bond is a value in the [0,1] interval. A high value means a highly correlated itemset. Note that single items have a bond of 1 by default.

The GRiST data was run through the CORI algorithm to find rare itemsets. The maxsupport was set at 0.8 and the patterns were extracted for various levels of BOND. Because most of the relationships are rare in the GRiST dataset, so a maximum support of 0.4 to 1 yields the same results.

Table 60 High risk prediction with CORI

BOND	No of rules	High recall	High precision	F-score
0.2	29	98.29	20.96	34.56
0.25	17	97.43	23.71	38.13
0.3	5	83.19	45.56	58.87
0.35	3	80.38	47.49	59.71
0.375	2	74.71	51.49	60.97
0.4	1	70.97	51.15	59.45

At this stage we are interested in identifying only high risk category patients (suicide risk>=5). By using the pattern extracted from the CORI algorithm, we achieve a maximum f-score of 60.097% but the precision score was only 51.49%. Using the high BOND value increases precision but our multi-rule approach is much more flexible. Next, we have tried to extract rules with high confidence by using the TopK Rules algorithm.

7.5.2 Using the TopKRules

The top-k association rules are the k most frequent association rules in the database having a confidence higher or equal to minimum confidence (Fournier-Viger et al.,

2016b). Other association rules mining algorithms requires us to set a minimum support (minsup) parameter, this is hard to set (users usually set it by trial and error, which is very time consuming). TopKRules solves this problem by letting users directly indicate k, the number of rules to be discovered instead of using minsup (Fournier-Viger et al., 2016b).

It provides the benefit of being very intuitive to use. It should be noted that the problem of top-k association rule mining is more computationally expensive than the problem of association rule mining. Using TopKRules is recommended for k values of up to 5000, depending on the datasets (Fournier-Viger, Wu, & Tseng, 2012).

TopKRules takes three parameters as input:

1. a transaction database,
2. a parameter k representing the number of association rules to be discovered (a positive integer),
3. a parameter minconf representing the minimum confidence that the association rules should have (a value in [0,1] representing a percentage).

The GRiST dataset was used to find the top 1000 association rules. We have run the algorithm with various levels of confidence. Then the extracted rules were used to predict suicide risk.

Table 61 High risk prediction with Top K rules

Confidence	Rule count	Min rules	High recall	High precession	F-score
0.8	36	1	54.54	66.52	59.95
		2	48.33	71.08	57.53
		3	42.64	74.87	54.33
0.85	27	1	43.16	72.55	54.13
		2	37.35	79.55	50.84
		3	33.42	82.85	47.63
0.9	6	1	26.26	86.34	40.27
		2	23.02	89.29	36.61
		3	20.91	90.54	33.97

From the above table we can see that the precision of high risk prediction can be up to 90.54%. Recall decreases as the precision increases. Only a few other algorithms such as AprioriRare and MNR (minimal non-redundant association rules) were also experimented with. They have produced similar but slightly less favourable results. One of the benefits observed with the TopKRules algorithm is that it is simple and produces a wide range of precision results.

7.6 The Multi-rule Risk Prediction Method

Based on the experimental results and the above discussion, we propose an algorithmic procedure for Risk Prediction from Node Association rules. The proposed method can be defined as below:

Method: Predict Patient Risk Level by Association Rules

```
1  Make list of Association Rules
2  Extract Patient attributes
3  minmatchcount=4;
4  matchcount=0;
5  foreach Rules do
6  |  if attributes match with Rule then
7  |  |  matchcount=matchcount+1;
8  |  |  break;
9  |  end
10 end
11 if matchcount ≥ minmatchcount then
12 |  high risk category
13 else
14 |  low risk category
15 end
```

Figure 20 The Multi-rule risk prediction method

The proposed algorithm can be adjusted by the confidence and support parameters of the FP-Growth association rules learning algorithm. The output can be dynamically adjusted as per the required accuracy by changing these parameters. The rules can also be extracted by other methods such as TopKRules. As the association rules can be learned offline, therefore it could be used in a real time environment where fast processing is required. It can predict high risk patients with up to 90% precision.

By using the proposed method, we can add an alert mechanism in the GRiST CDSS system work flow. The alert can be given as soon as any matching rules are found even before the completion of the assessment. If regression analysis is used for prediction then we have to wait for the assessment to be fully completed before any alert can be given. The following diagram shows the modified GRiST CDSS system.

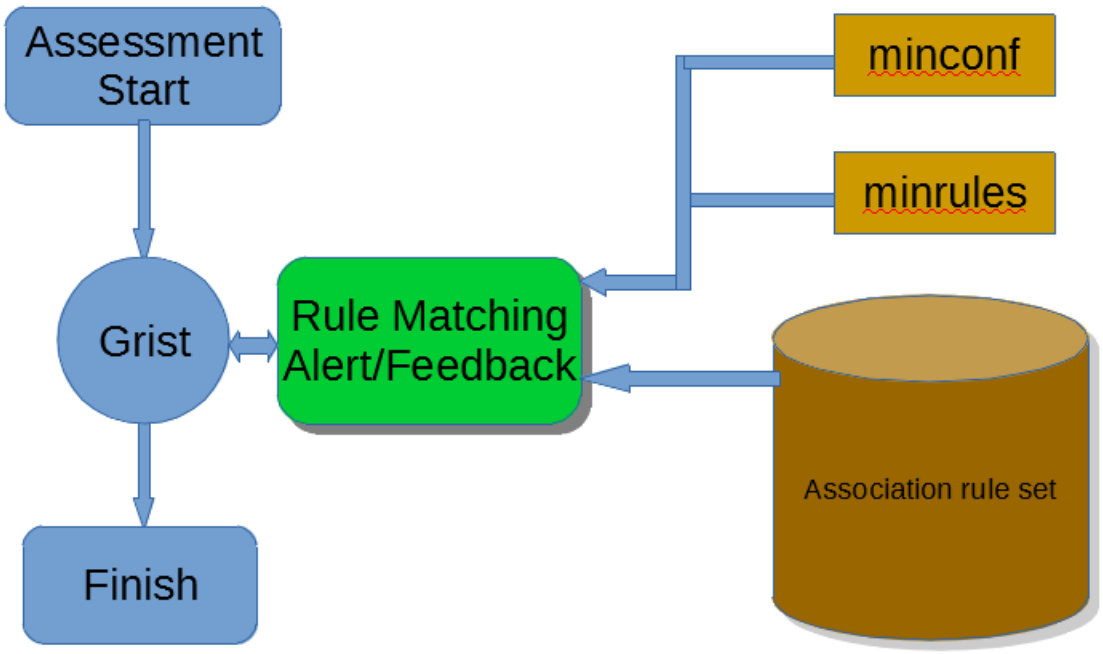

Figure 21 Proposed alerting mechanism for GRiST CDSS

7.7 Summary

There exist some statistically significant relationships between the GRiST nodes and they were confirmed by both phrase and numerical analysis. It has been found that the presence of a greater number of node relationships in an assessment indicates a potentially higher risk of suicide. This is a new finding in the context of the GRiST dataset and might help us to further analyse the GRIST data. The presence of node relationships that produce higher average risk rather than the individual node's average risk is considered as a high-risk relationship.

To further investigate the relationships among nodes we have used association rule mining techniques. Within the GRiST dataset, the higher risk of suicide is a rare event. A new multi-rules based method has been proposed to find the high risk category patients dynamically at the time of assessment. The proposed method is adjustable based on the expected accuracy of the prediction. Empirical data shows that the proposed method can be used to predict high risk patients with greater accuracy (up to 90%) than the regression method (66%). Another benefit of this method is that we can predict before the completion of the assessment and as soon as the patterns are detected. This is a significant contribution to the GRiST project. This fulfils one of our key objectives, which was to find high risk category patients with better precision.

Because a disease is generally a rare event in a dataset, hence low support is used in association rule mining in biomedical literature. We have compared many rare event mining techniques found in the literature. Empirical results show that our proposed multi-rule approach works better than the other method such as CORI or TopKRules itemset mining alone.

After completion of the assessment, clinicians provide their own risk judgement. To make the GRiST system a more intelligent CDSS, we would like to assess the reliability of the clinician's judgement and provide feedback accordingly. In the following chapter, we discuss the differences between the calculated/predicted risk and the clinicians given risk and propose a novel method to improve the reliability of the risk judgement.

8 Reliability of Risk Judgement

8.1 Introduction

The use of clinical decision support systems (CDSSs) has increased recently and has shown an improvement in productivity, reduction of medication errors and an increase in quality of hospital services (Al-gamdi, 2014). However, the acceptance of CDSSs is hampered by the complexity of the system, their time-consuming nature, and a general lack of accurate decision support (Al-gamdi, 2014). In this chapter, we address the decision accuracy problem in the context of the GRiST decision support system.

One of the aims of this research was to explain the differences between the clinician given risk and the calculated risk. The calculated risk may come from regression or any other machine learning approach. Presently, when the clinician provides a risk judgement, there is no way to validate this judgement. One way to validate this could be to compare the clinician given risk with the risk calculated by regression analysis. We may consider the calculated risk as the consensus risk. When there is a difference between the clinician and calculated risk, it would be extremely valuable to be able to explain that difference. The CDSS can then alert the clinician and point to the possible improvement strategy.

To solve the above mentioned problem, we have chosen to use the Information theoretic approach. For clustering comparison information theoretic measures have been employed because of their strong mathematical foundation, and ability to detect non-linear similarities (Vinh, Epps, & Bailey, 2010). We might use the information gain indirectly as to explain why the clinician's judgement and calculated risk differ, especially when the difference is high. The assumption is that the difference between the clinicians and calculated risk increases when the overall information gain of the assessment is low. In other words, the assessment was probably not conducted encompassing all the aspects of the evaluation process, which resulted in less information gain and more differences.

Hypothesis 1: The accuracy of the risk judgement depends on the information collected by the clinician at the time of the risk judgement and a higher level of risk requires more information collection.

Hypothesis 2: The difference between the clinicians and the predicted risk is inversely correlated with the total information gain of the assessment.

Hypothesis 3: The clinician given risk can be adjusted to bring it closer to the consensus risk (calculated risk) based on the total information gain of the assessment.

8.2 **Dataset**

For this experiment, the same dataset from the GRiST database was chosen, which was used for regression analysis in Chapter 6. GRiST has more than 436 nodes and it is regularly updated with new assessment types. A subset of 141 nodes were chosen, which have a scale data type. The reason for this is to keep the number of nodes computationally manageable and regression analysis was done using these nodes. We wanted to compare the risk from the regression analysis with the clinicians given risk and explain the differences with information gain.

A total of 46903 assessments were chosen from the GRiST database. These assessments had at least 1KB of text comments and suicide risk judgement was more than zero. The other assessments, which were not suicide related, were eliminated. Some of them were only related to self-harm or absconding. The following table shows the distribution of assessments across the different suicide risk categories.

Table 62 GRiST data distribution across risk levels

Suicide risk	No of assessments
1	13480
2	11490
3	8539
4	4688
5	3793
6	1707
7	1608
8	1077
9	382
10	139
Total	46903

The following table shows the number of assessments based on when they were conducted. Assessments conducted between 2011 and 2013 were used for training and the rest were used for testing.

Table 63 Test and Training dataset from GRiST

Assessment year	No of Assessment	Used for
2011	4663	training
2012	6301	training
2013	10238	training
2014	12747	test
2015	12953	test

The multiple linear regression analysis was done on the dataset using the 141 scale type nodes as attributes and suicide risk was calculated. The 10-fold validation correlation coefficient was 0.78. The calculated suicide risk was considered as consensus suicide risk for further analysis.

The following table shows the correlation coefficient between the calculated and predicted risk across varying risk levels. The correlation co-efficient for the high risk category patients was much lower. It indicates that the clinicians given risk differs significantly from calculated risk for the high risk category patients.

Table 64 Correlation between clinical and calculated risk

Risk Level	Correlation
>0	0.785
>1	0.726
>2	0.661
>3	0.590
>4	0.520
>5	0.421
>6	0.341
>7	0.252

From the data, we can see that the overall correlation is 0.78 (between the clinicians and calculated risk), but the predictive accuracy varies as the level of risk goes higher.
Our objective was to find methods to identify probable inaccurate risk judgement and notify clinicians in real time. For example, the system may notify the clinician, that the given risk level requires more information collection.

We have applied multiple methods to achieve our objective. Sequentially, the later described methods address some of the limitations of the prior methods. The applied methods and their respective results are described in the following sections.

8.3 Method A: Information Gain

8.3.1 Introduction to Entropy

Information theory and its use in communication was originally proposed by Shannon (1948) in his seminal paper “A mathematical theory of communication”. Most of the ideas came from this paper.

Information we get from observing the occurrence of an event having probability p is defined as (Carter, 2007), (Shannon, 1948):

$$I(p) = -\log_b(p) \tag{15}$$

Where:

p = probability

b = base (base 2 is used in information theory)

“The entropy of a probability distribution is just the expected value of the information of the distribution” (Carter, 2007, p. 25). It is calculated as the weighted average amount of the information from the event.

Suppose X is a discrete random variable which takes values from the set X = {x1,x2, ...,xn}, and is defined by a probability distribution p(X), then the entropy of the random variable can be defined by the following equation (McEliece, 2013), (Thomas M Cover & Thomas, 2005) and (Shannon, 1948):

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_b p(x_i) \tag{16}$$

If the log in the above equation is based on 2 then the entropy is expressed in bits and If the log is based on natural log, then the entropy is expressed in nats (McEliece, 2013).

In computing, entropy is commonly expressed in bits, and unless otherwise stated, we will assume a logarithm with base 2.

The concept of entropy is related to ‘uncertainty’, ‘randomness’ or ‘noise’ in a system. High Entropy means that the probability distribution is uniform. There is an equal chance of obtaining any possible value. A Low Entropy means that the distribution is not uniform. Hence, it is more predictable.

In a tree-type structure if a choice is broken down into two successive choices, the original H should be the weighted sum of the individual values of H (Shannon, 1948). The meaning of this is illustrated in Figure 22 reproduced from (Shannon, 1948).

Figure 22 Decomposition of a choice from three possibilities reproduced from (Shannon,1948)

The left tree has three branches with p1=1/2, p2=1/3, p3=1/6. The right has two branch with a probability of 1/2, and the second branch makes more branches as shown in the second image. According to Shannon (1948) the final results have the same probabilities for both of these trees. Hence,

$$H(1/2,1/3,1/6)=H(1/2,1/2) +(1/2)\ H(2/3,1/3)$$

The coefficient 1/2 is because the second branch only occurs half the time (Shannon, 1948). This calculation technique is used to calculate information gain, which is

described next. More detail on information theory and entropy can be found in (T M Cover & Thomas, 2012), (McEliece, 2013), (Carter, 2007) and (MacKay, 2003).

8.3.2 Information Gain (IG)

Information gain (IG) is a measure of the reduction of uncertainty in class prediction, if the only information available is the presence of a feature and the corresponding class distribution (Roobaert, Karakoulas, & Chawla, 2006). It measures the expected reduction in entropy by partitioning a group according to a feature or the attribute in question (Bahety, 2014).

For example if we have a data set T which has C1,C2,…Ck class. |T| is total number of class and |Ck| is the number of items in k class. The total entropy or information of the set T is:

$$Info(T) = -\sum_{i=1}^{k}(|\mathrm{Ci}|)/|T|) * log_2(|\mathrm{Ci}|/|T|)) \quad (17)$$

Considering the similar measurement after T has been partitioned by attribute X into T1, T2, … Tn subsets. We can calculate the entropy of each of these subsets by using the above formula. The expected information of the attribute X is the weighted sum of each subsets (Kantardzic, 2011).

$$Info_x(T) = \sum_{i=1}^{n}((|T_i|/|T|) * Info(T_i)) \quad (18)$$

Then the loss of entropy is the information gain for the attribute X. We get

$$Gain(X) = Info(T) - Info_x(T) \quad (19)$$

In decision tree algorithms the attribute that maximises the Gain(X), is selected for tree split. The above explanation is summarised from (Kantardzic, 2011). This calculation technique is used in this research.

8.3.3 Gain Ratio

Decision tree learning algorithms (e.g. ID3) uses information gain to select the best node for split. “One limitation of ID3 is that it is overly sensitive to features with large numbers of values” (Hssina, Merbouha, Ezzikouri, & Erritali, 2014, p. 15). To overcome this problem Gain(X) is normalised using SplitInfo(X) as described in (Kantardzic, 2011).

$$SplitInfo(X) = -\sum_{i=1}^{n}\left(\frac{|Ti|}{|T|}\right) log_2(\frac{|Ti|}{|T|}) \quad (20)$$

This represents the potential information generated by dividing set T into n subsets T_i. A new gain measure is defined (Kantardzic, 2011) as:

$$\text{Gain-Ratio(X)=Gain(X) /Split-info(X)} \quad (21)$$

This measure is robust and typically gives a consistently better choice of a test than the previous gain measure used (Kantardzic, 2011). Several splitting schemes have been compared in the past (Banfield, Hall, Bowyer, & Kegelmeyer, 2007). We have used both gain and gain ratio in our experiments. The exact method detailing how it was done in this research is explained in the next section.

8.3.4 Example of IG Calculation

The information gain of each node can be calculated based on the theoretical discussion above and the examples described in (Du, Du, Zhan, & Zhan, 2002) and (Kantardzic,

2011). It might be better to explain the method with an example. This example is adapted from (Kantardzic, 2011) and (Amro, 2009).

Let us consider a single node of GRiST ontology such as [depression] and run a calculation based on the information depicted in the figure below.

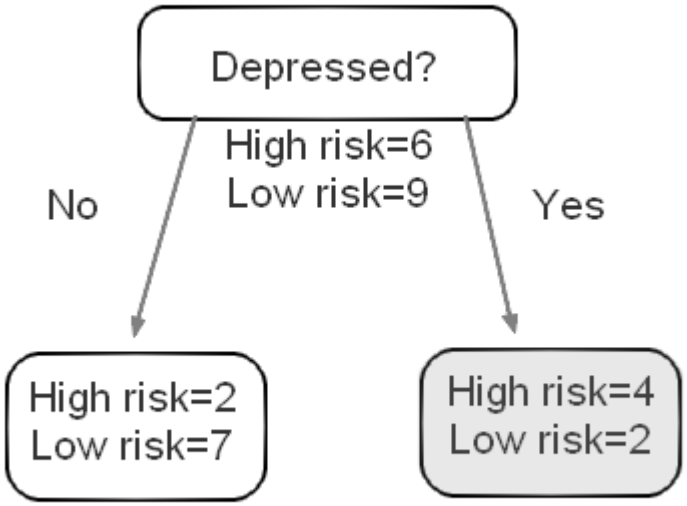

Figure 23 Information gain calculation example

Suppose we have 15 patients, of which 6 are of high suicide risk category and 9 are of low suicide risk category. This gives us the probability of high risk, P(H)=6/15 and a probability of low risk, P(L)=9/15.

Using the formula to calculate entropy we get:

Entropy_before = - (6/15)*$\log_2$(6/15) - (9/15)*$\log_2$(9/15) = 0.9709

Based on the answer to the depression status question, we can divide the patients into two groups. For the group on the left (depression=NO) we get the entropy as follows:

Entropy_NO = - (2/9)*log2(2/9) - (7/9)*log2(7/9) = 0.7642

And for the group on the right (depression=YES), we get the entropy:

Entropy_YES = - (4/6)*log2(4/6) - (2/6)*log2(2/6) = 0.9182

Now we can combine these two entropy's as a weighted sum, which gives us the entropy after the split:

Entropy_after = 9/15*Entropy_NO + 6/15*Entropy_YES = 0.8259

If we deduct entropy_after from the entropy_before we get the information gain of the depression node.

Information_Gain = Entropy_before - Entropy_after = 0.1449

We can say that after asking the 'depression question' we have gained some information about the patient's suicide risk. The information gain amount is 0.1449. Like this example, information gain can be calculated for any of the GRiST nodes.

8.3.5 Gain Ratio Calculation

To calculate the gain ratio of each of the GRiST nodes I have used the machine-learning tool Weka (Hall et al., 2009) that implements various attribute selection algorithms including GainRatio. The GainRatio attribute selection option was used to calculate gainratio of each of the selected GRiST nodes. The gain ratio calculated by myself using the method described above and by Weka have both produced the same results. Subsequently for simplicity purposes, Weka was used. The following table shows some of the results.

Table 65 Gain Ratio of GRiST nodes

GRiST node	Description	Gain Ratio
suic_plan_real_answer	Realism of the suicide plan	0.19787
suic_id_hi_risk_answer	High risk suicide ideation	0.1937
suic_steps_takn_answer	Suicide steps taken	0.19063
suic_prosp_leth_answer	Lethality of the methods	0.1889
suic_eol_prep_answer	End of life preparation	0.17803
suic_pot_trig_answer	Potential trigger	0.17285
suic_id_strngth_answer	Suicide ideation intensity	0.16096

The calculated Gain ratio was used for subsequent analysis and application. The Gain ratio was calculated from the training dataset and applied for risk adjustment in the test dataset. Later in this chapter, the term information gain is sometimes used in general to refer to gain ratio or the total information collected by an assessment.

8.3.6 Experimental Results and Analysis

The relationship between a node and suicide risk can be viewed in the way the node correlates with suicide risk. A node's correlation with suicide risk may also be related to the information gain of the node. For further exploration a node's information gain and its correlation with suicide risk were compared and some of the results are shown below.

Table 66 Gain Ratio of GRiST nodes

GRiST node	Description	Correlation with suicide	Gain ratio
suic_plan_real_answer	Realism of the suicide plan	0.34	0.19787
suic_id_hi_risk_answer	High risk suicide ideation	0.80	0.1937
suic_steps_takn_answer	Suicide steps taken	0.31	0.19063
suic_prosp_leth_answer	Lethality of the methods	0.21	0.1889
suic_eol_prep_answer	End of life preparation	0.23	0.17803
suic_pot_trig_answer	Potential trigger	0.61	0.17285
suic_id_strngth_answer	Suicide ideation intensity	0.75	0.16096

Please note: The meaning of the GRiST node can be found in Appendix A.

The analysis reveals that a node having a high correlation with suicide risk does not always mean that the node would provide high information gain (gain ratio). For example, sh_steps_taken (steps taken) has correlation 0.31 but information gain is 0.190 and suic_pot_trig (potential trigger) has a correlation of 0.61 but information gain is 0.172. Which may be interpreted as the knowledge of 'steps taken' provides us with more of an indication than the knowledge of a 'potential trigger' to identify the suicide risk.

We can assume that if a patient came for an assessment and a high-risk score was given then the patient should have been assessed more rigorously (Hypothesis 1). In other words, many relevant questions should have been asked before making the judgement hence the total information gain should be high. On the other hand, when a low risk score was given then there was possibly a lack of evidence or only a few questions were needed to be asked.

For a particular assessment, we can calculate "total gain" by adding the gain of each of the answered questions. Technically, this is not the total information gain as there may be interactivity between nodes. Nevertheless, we use the "total gain" as a relative measure for total information collected by an assessment. It is the total number of questions asked weighted by information gain/ gain ratio. The following is the formula to calculate sum total gain.

Sum total gain= question1 * gainratio1 + question2 * gainratio2 + …….

Strictly speaking, this is not the total gain of the assessment because there may be mutual interdependency among some questions. Acknowledging this limitation, we assume this could be considered a good indicator of the total information gain of the assessment for comparison between two assessments. We have addressed this limitation with other methods described later in this chapter.

Table 67 Gain ratio and average risk difference

Suicide risk	Sum total Gain Ratio	Average risk difference
1	0.44	-0.77
2	0.66	-0.20

Suicide risk	Sum total Gain Ratio	Average risk difference
3	0.89	-0.19
4	1.13	0.47
5	1.26	0.87
6	1.49	1.08
7	1.67	1.39
8	1.80	1.63
9	1.92	1.88
10	1.80	2.25

The above table shows the average total gain ratio and the risk difference per risk category. We can see that as the risk increased more information was collected. In other words, with increasing risk, perhaps more information was available to collect. The trend of this result was expected. We can also observe that the average risk difference is higher in the high-risk category patients. For suicide risk level 10, we can see that the gain is slightly less than the gain of risk level 9. A possible reason could be that in very high-risk situations there could possibly be other factors influencing a clinician's risk judgement. The following figure shows the total gain and risk level trend graphically.

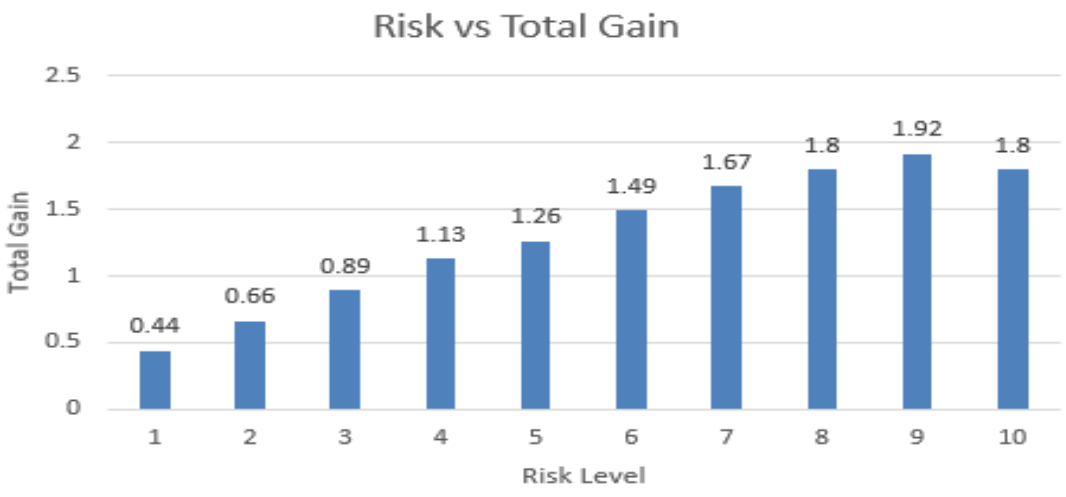

Figure 24 Sum total gain for different risk level

If an assessment has been done without asking all the questions, then ideally, we could expect an anomaly. We can calculate the difference between the clinical judgement and

calculated risk for each of the assessments. We assume that these differences would be inversely proportional to the total gain. This would be especially true for high risk category patients, as they may require more information collection. The following table shows the correlation between risk difference and the total information gain of the assessment and this helps to prove our first hypothesis.

Table 68 Total gain vs Risk difference

Risk level	Risk Difference vs Total gain
>1	-0.267
>2	-0.435
>3	-0.554
>4	-0.640
>5	-0.731
>6	-0.781
>7	-0.809

Hypothesis 2: The difference between the clinicians and the predicted risk is inversely correlated with the total information gain of the assessment.

The data in the above table shows that total gain is inversely proportional to the risk difference. The assessment, which has more total information gain, matches more closely with the consensus risk. In other words, the difference between the clinicians and the predicted risk is inversely correlated with the total information gain of the assessment. It is an important finding from this dataset.

From the data, we also see that the risk difference is generally higher when the clinician's given risk is high. We also know that the correlation between the clinician's risk and the predicted risk is lower for higher risk patients. These results suggest that there may be a potential link between information gain and risk assessment accuracy. One of the objectives of this research was to find potential reasons why the clinicians and calculated risk differ.

For example, for all the assessments the correlation between the clinicians and calculated risk is approximately 0.78 and for assessments where the suicide risk is greater than 5, the correlation is only 0.42. In other words, the prediction of high risk is not as accurate as the prediction of low risk. It is desirable to predict higher risk cases more accurately. We assume that total information gain might help us to adjust the clinicians given risk, especially when the risk is in higher categories (>=5). This leads to our third hypothesis of this chapter, which is discussed in the following section.

8.3.7 Adjustment of the Clinical Judgement

We assume that we can adjust the clinical judgement based on the total gain of the assessment. We can notify the clinician interactively to suggest collecting more information based on the clinicians given risk level and the total gain of the assessment.

Hypothesis 3: We can adjust the clinicians given risk based on the information gain and make it closer to the consensus risk (calculated risk).

We can adjust the clinical judgement based on the total gain and achieve a risk judgement that is much closer to the calculated risk. We know from the empirical data that the risk difference is higher in high risk category patients and the sum total information gain is inversely proportional to the risk difference. This allows us to adjust the clinicians given risk based on total information gain. I have taken the following steps to adjust the clinicians given risk:

Step 1: The linear regression equation between Risk Difference and total Gain Ratio was calculated.

$$\text{Risk Difference} = \text{Gain Ratio} * -1.55 + 3.8 \quad (22)$$

The above equation was created from the data in which the clinicians given risk was greater than 5. Please note that it is not a static equation. It can change based on the risk level.

Step 2: Calculate the total Gain Ratio of an assessment and apply step 1 to find out risk differences. This is the total risk adjustment we need to make.

Step 3: Apply the calculated risk difference to the clinicians given risk.

Step 4: Correlate this new risk with the risk calculated by regression.

The following table shows the correlation with the calculated risk before and after the adjustment.

Table 69 Risk adjustment by information gain

Risk level	Correlation before	Correlation after
>1	0.726	0.780
>2	0.661	0.777
>3	0.590	0.779
>4	0.520	0.781
>5	0.421	0.794
>6	0.341	0.811
>7	0.252	0.823

From the above table we see that if we adjust the clinical judgement then the correlation between the adjusted clinical and calculated risk improves significantly. For the clinician given risk>5 the correlation between the clinicians and the calculated risk was 0.42 and after adjustments it became 0.79. The similar pattern (before adjustments the correlation =0.33 and after adjustment the correlation =0.60) was observed with the risk prediction data provided by other GRiST researchers (Nagy, 2016). The adjustments significantly improved the correlation. Some sample data are shown in the table below.

Table 70 Adjusted risk example

Suicide risk	Adjustment	Adjusted risk	Total relative weight	Predicted risk
6	-1.78	4.22	0.30135	4.115
6	+0.19	6.19	0.50236	6.102
6	-3.57	2.43	0.11919	2.631
6	-1.11	4.89	0.36924	5.309
6	+1.01	7.01	0.58553	7.881
7	-0.99	6.01	0.38164	4.407
7	-2.33	4.67	0.24517	5.793
7	-1.44	5.56	0.33568	5.598
7	+0.81	7.81	0.56491	6.826
7	+0.96	7.96	0.58081	8.362
8	-1.13	6.87	0.36727	5.5
8	-1.15	6.85	0.36528	6.606
8	+0.58	8.58	0.54173	9.094
8	+0.05	8.05	0.48767	8.278
8	-3.29	4.71	0.14774	3.467
9	+0.22	9.22	0.50496	9.498
9	-2.49	6.51	0.22891	5.177
9	-1.81	7.19	0.2989	6.384
9	-2.52	6.48	0.22637	4.421
9	+0.89	9.89	0.57375	9.875
10	-3.13	6.87	0.16422	5.094
10	-1.72	8.28	0.30748	6.696
10	-3.14	6.86	0.16306	3.768
10	+0.66	10.66	0.55011	10.97
10	+0.53	10.53	0.5368	9.171

The main assumption here is that if an assessment has only low total information gain then the clinicians probably did not undertake the full assessment or did not ask the most critical (high gain) questions. In this scenario, it is more likely the given risk was not accurate. If the gain is low, then we might adjust this assessment before we compare it with the calculated risk. Please note that we are considering the calculated risk as a consensus risk.

The main idea can be explained by a hypothetical example. If a doctor diagnoses somebody with cancer without performing multiple tests, then we might say the

diagnosis was not comprehensive. The information gain would be too low in this case. We expect the accuracy of the judgement to be higher when the overall information gain is high and vice versa.

If we adjust the clinicians given risk, then the correlation between the clinician and the consensus risk improves significantly (from 0.42 to 0.79 for risk>5). An explanation for this could be that the calculated risks came from mathematical data, but the clinicians given risk came from a subjective judgement. That is why an adjustment may need to be done on the clinicians given risk. In this case, the calculated risk is also considered as a consensus risk, so it should not be adjusted.

We have used regression generated risk as consensus risk. It could be calculated by any other method. Because the data size was approximately 50,000 so we can reasonably assume that risk calculated by regression is suitable to be considered as a consensus risk. I have also compared the data with the risks predicted by different methods by other members of the GRiST team such as by Nagy (2016). To overcome the problem of inter node correlation affecting the total information gain; we have used other different methods, which are described in the following sections.

8.4 Method B: R-Square (variance) Analysis

Multiple linear regression analysis is widely used in many scientific fields, including public health, to evaluate how an outcome or response variable is related to a set of predictors (Chao, Zhao, Kupper, & Nylander-French, 2008). One is generally interested in finding the relative contribution of each predictor towards explaining variance in the criterion variable. It becomes difficult when the predictor variables are typically correlated with one another.

“Relative importance” refers to the quantification of an individual predictor’s contribution to a multiple regression model. Assessment of relative importance in linear models is simple, as long as all regressors are uncorrelated: Each regressor’s contribution is

simply the R-square from univariate regression, and all univariate R-square-values add up to the full models' R-square (Grömping, 2006).

A sequential analysis might circumvent this problem, but in most cases, there is no obvious way in which order the predictors should be considered (Nimon & Oswald, 2013). There are now procedures available by which we can partition the R-square into pseudo-orthogonal portions, each portion representing the relative contribution of one predictor variable. "Relative weights" is a way to partition an MLR model R-square across predictors. They are computed by first transforming p predictors into a new set of p variables that are uncorrelated with one another, yet are correlated as highly as possible with the original predictors (Nimon & Oswald, 2013).

8.4.1 Calculation of Relative Weights

There is an R-package called "yhat", which implements the relative weight calculation method. We have used the 'yhat' package from R to calculate relative weights of each individual node from the test data. Source of the 'Yhat' R package and documentation is available from the URL (https://CRAN.R-project.org/package=yhat).

Table 71 Relative weights of GRiST sample nodes

GRiST Node Name	Description	Relative weights
suic_pot_trig_answer	Potential trigger	0.08493
suic_id_hi_risk_answer	High risk suicide ideation	0.04922
suic_p_trig_mtch_answer	Trigger and past attempt match	0.03516
suic_id_strngth_answer	Suicide ideation strength	0.0344
suic_id_control_answer	Suicide ideation control	0.03222
suic_regret_answer	Suicide regret	0.02481
gen_life_not_livng_answer	Life not worth living	0.02388
suic_eol_prep_answer	End of life preperation	0.02184
suic_ser_succd_answer	Seriousness of success	0.02078
suic_lethality_answer	Suicide lethality	0.01982

Please note: The meaning of the GRiST node can be found in Appendix A.

8.4.2 Relative Weight for Reliability Analysis

We can calculate the total sum of R-square of an assessment from each individual node's relative weight. Theoretically, this total sum of R-square means how much of the variance of the data is explained by the clinical assessment. We can assume that the bigger sum would indicate thorough assessment hence would produce better clinical judgements. In other words, the risk difference would be inversely proportional to the total sum of R-square.

Table 72 Correlation between risk difference and the total sum of R-square

Risk level	Risk Difference vs Sum of R-Square
>1	-0.314
>2	-0.499
>3	-0.620
>4	-0.700
>5	-0.783
>6	-0.827
>7	-0.850

The above table shows that indeed for high risk patients the risk difference is inversely correlated with the total sum of R-square. For each level of risk, we have calculated the adjustments required from the training data as we have done previously with total information gain. Then applied the adjustments to the data collected in both 2014 and 2015.

Firstly, we have calculated the total sum of R-square and used that to predict the risk difference for training data and build a regression model. Then we applied the model to test data to adjust the clinical judgement based on the total sum of R-square.

Table 73 Adjusted clinical risk by total sum of R-square

Risk level	Correlation before	Correlation after
>1	0.726	0.790
>2	0.661	0.798
>3	0.590	0.810
>4	0.520	0.814
>5	0.421	0.832
>6	0.341	0.851
>7	0.252	0.861

From the above table we see that if we adjust the clinical judgement with the total sum of R-square then the correlation between the clinical and calculated risk improves significantly.

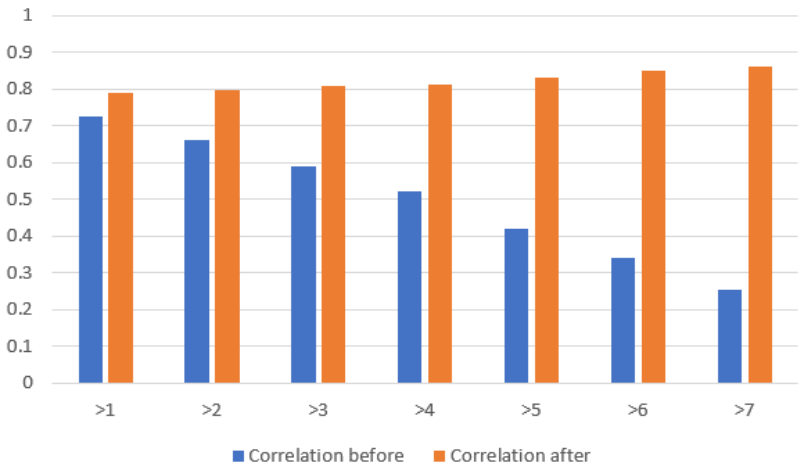

Figure 25 The Improvement of risk judgement

The above graph shows the correlation before and after the adjustment of the clinical judgement. The adjustment with relative weight produced better results than the information gain.

8.5 Method C: Using Number of Questions

We could simply use the number of questions that has been asked and adjust the clinician's given risk. This is the baseline method. Comparison with this method could highlight the utility of the other methods.

Table 74 Correlation between No of Questions and risk difference

Risk level	Risk Difference vs No of questions
>1	-0.179
>2	-0.295
>3	-0.375
>4	-0.420
>5	-0.469
>6	-0.498
>7	-0.503

The above table shows, that indeed for high risk patients the risk difference is inversely correlated with the total number of questions asked. However, the effect is much less than the Information gain and R-square (relative weight) method.

Table 75 Risk adjustment by No of questions

Risk level	Correlation before	Correlation after
>1	0.726	0.748
>2	0.661	0.711
>3	0.590	0.675
>4	0.520	0.641
>5	0.421	0.598
>6	0.341	0.575
>7	0.252	0.545

From the above table, we can see that the adjustment done using a simple number of questions asked can produce a better correlation between the adjusted clinicians and

calculated risk. Adjustments done with Information gain or R-square (relative weight) provides much better results than the simple number of questions.

8.6 Comparison of Different Methods

Using the number of questions weighted by information gain provides better results than just the simple number of questions. If we use the sum of Relative weights of the questions we obtain much better results. The following table shows the comparison amongst the different approaches.

Table 76 Comparisons amongst the different approaches

Risk level	Correlation before	No of question	Information gain	Relative weight
>1	0.726	0.748	0.780	0.790
>2	0.661	0.711	0.777	0.798
>3	0.590	0.675	0.779	0.810
>4	0.520	0.641	0.781	0.814
>5	0.421	0.598	0.794	0.832
>6	0.341	0.575	0.811	0.851
>7	0.252	0.545	0.823	0.861

Information gain provides better results than the simple total number of questions because in the total gain calculation we take into account the relative importance of the questions. Because of the possible interaction between GRiST nodes calculating total gain of the assessment is difficult. Relative weight performs better than total gain probably because we eliminated the interdependency of the nodes. Two correlated predictive variables would not affect the overall total relative weight because the algorithm firstly converts them to orthogonal variables (Nimon & Oswald, 2013). The following graph shows the correlation between the adjusted and calculated suicide risk at the different risk levels.

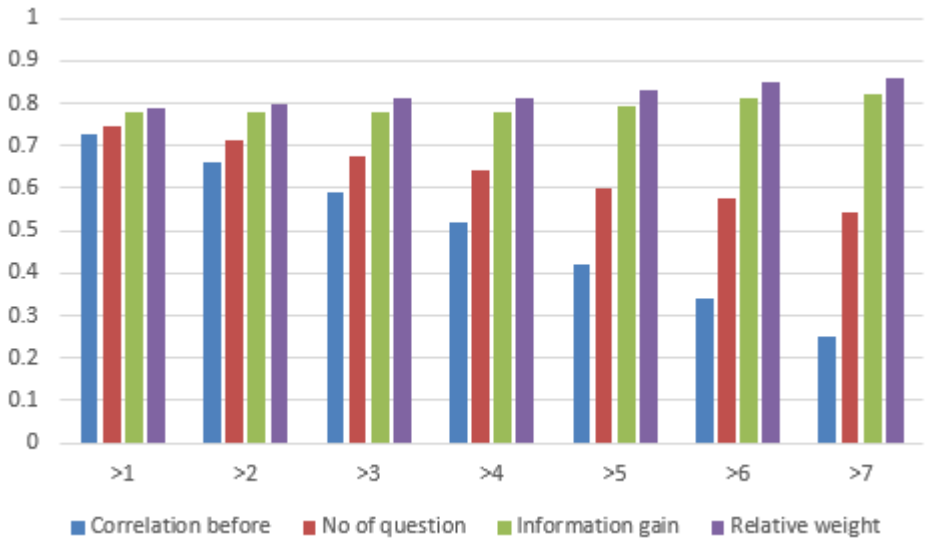

Figure 26 Improvement of assessment accuracy by different methods

8.7 The Reliability Assessment Method

We are proposing a novel method to check the reliability of the clinical judgement. If the total gain (or relative weight) of the assessment is less than the required gain (or weight) for that level of risk, then we might adjust the clinical judgement. Experimental data suggests that applying this adjustment brings the clinical judgement closer to the consensus risk. Similar trends have been observed with the risk calculated by regression analysis and the risk calculated by other methods and by other members of the GRiST team (Nagy, 2016).

The empirical data suggests that if implemented properly within a CDSS then the proposed method could guide a clinician towards the direction, where clinical judgement would more closely relate to the consensus. The system could alert the clinician to collect more data based on the risk levels and total information gain and that might all in all lead to a better assessment.

The following figure shows a typical possible implementation of the method. We can use information gain or relative weight to measure total information collected.

Method: Reliability of Risk Assessment

```
1  Calculate Information Gain of each node
2  Calculate Average Gain per Risk Level
3  Extract Patient attributes
4  totalgain=0;
5  foreach attributes do
6  |  totalgain=totalgain+attribute*informationgain of the attribute;
7  end
8  if totalgain ≥ averagegain of the risk level then
9  |  assessment is reliable
10 else
11 |  assessment may not be reliable
12 end
```

Figure 27 Reliability of the Risk Assessment

8.8 Applications of the Method

The following are some suggested applications of the method described in this chapter:

Application 1: This method can validate the GRiST assessments and explains the reason for the difference between the clinicians and the calculated risk. It was one of the main objectives of the research. Empirical data shows that a lack of Information gain could be a reason for the difference.

Application 2: Information gain tells us which nodes are important. We can use information gain as a node selection criterion. We can direct the clinician to the high gain nodes. We could redesign the assessment flow based on the Information gain.

Application 3: Adjusted risk can be used to trigger events for example prompting certain actions by the clinicians. Higher management can also filter out specific assessments based on information gain and risk level.

Application 4: Provide real-time notifications about the projected accuracy of the risk assessment. We may ask the clinicians to explore further if the given risk appears to be higher and the information gain is very low.

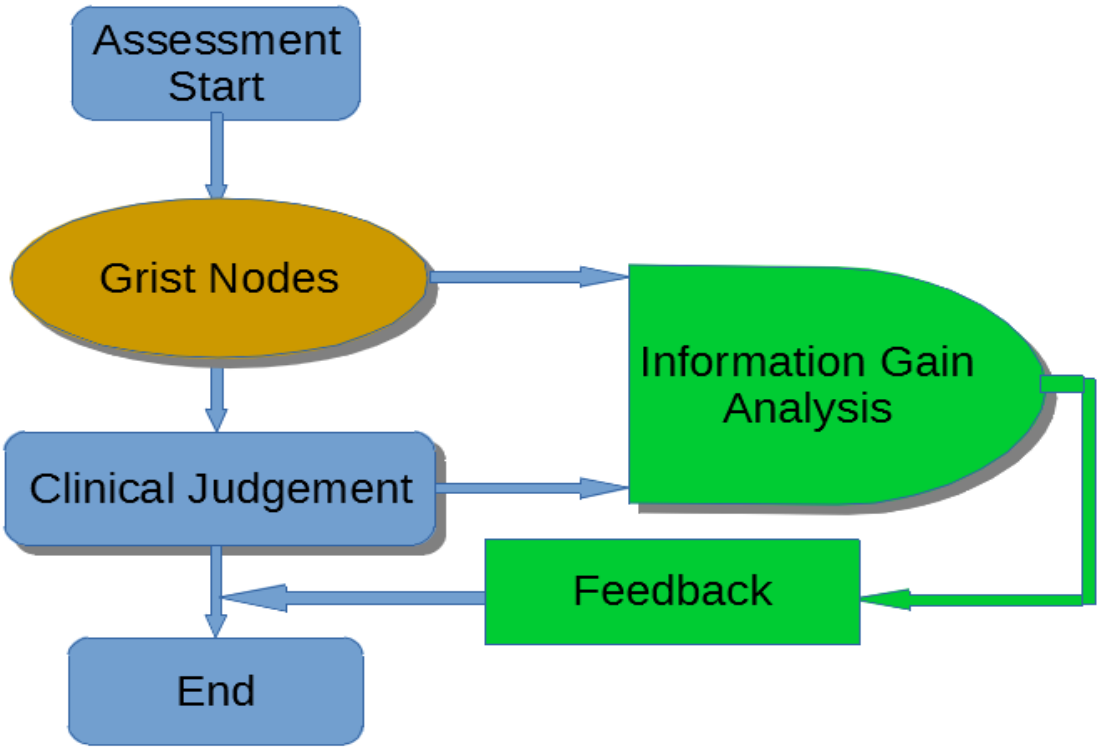

Figure 28 Application of Information Gain Analysis

8.9 **Summary**

It is desirable to be able to validate the risk judgement provided by the clinician automatically. We could calculate risk by using regression or other machine learning methods. Previously, we did not know why there was a difference between the clinician's given risk and the calculated risk. By using the proposed method, we could possibly analyse the clinical judgement based on the information collected before reaching that judgement and provide appropriate feedback.

The method can be applied in real-time, as the main calculation is not computationally expensive. Some of the calculations such as Gain calculation and regression equation calculation can be done periodically to bring them up-to-date with recent data. The method can be extended to other clinical decision support systems, which use a similar structure like the GRiST system.

The performance of the GRiST system can be greatly enhanced by adding background analysis capability. The ability to provide intelligent feedback might increase the acceptance of the system as suggested by (Al-gamdi, 2014). The sum total of information gain is calculated on the available data only, so any missing data would not affect this method. The missing data would reduce the total gain and may generate a warning.

To address the potential problem of inter node relationships; we have used ‘relative weights’ to calculate the total weight of the assessment. The proposed method also provides a facility to direct clinicians to collect important data that might help improve the overall accuracy of the judgement. This technique could be applied in real-time at the time of assessment to provide alerts or as a management tool to evaluate assessments at a later date. Low gain but high risk assessments could generate management alerts.

9 Conclusion

9.1 Introduction

According to the Mental Health Foundation (2015) one in four people in the UK will experience a mental health problem in any given year. It states that mental health problems are one of the main causes of the burden of disease worldwide. The context of this research was the GRiST clinical decision support system, which assesses mental health risks. I have tried to improve the interactivity of the GRiST system and validate the clinicians given risk score. Techniques starting from natural language processing, statistical analysis to information theoretic analysis have been investigated.

This chapter gives a concluding critical analysis of each of the activities and findings followed by suggestions for future works. The practical implications of this research on the GRiST system and on any other similar CDSS system are discussed ending with a concluding remark.

9.2 Empirical Findings

Empirical findings of these activities with concluding critical analysis are given in the following individual sections.

9.2.1 Concept Extraction

To use clinical narratives found in the CDSS firstly we need to extract relevant concepts from them. Many well-known phrase extraction algorithms have been applied and their performance was analysed. The outcome of the off the shelf phrase extraction algorithms was not very useful. Algorithms that did not require training data produced

many irrelevant phrases (low precision). On the other hand, algorithms that required training had low recall.

SNOMED-CT is a medical terminology database and there are tools available to extract phrases based on this ontology. Originally, Metamap was used and subsequently cTAKES was used to identify phrases in the comments that relate to SNOMED-CT and extracted them as important concepts. The results showed that cTAKES and Metamap could both extract phrases with high precision. But unfortunately, their recall was very low. The use of these tools would miss a lot of valuable information.

After reviewing the outcome of the other methods, we have developed a two-stage method that can extract concepts unsupervised. The proposed method first extracts phrases with linguistics patterns, in the second stage, it filters the extracted phrases for the domain relevancy using vector based semantic filtering. Domain relevancy is measured by calculating the semantic distance of a phrase to the list of frequently found concepts in that domain. The empirical results show that our method can achieve better than other methods tried on the i2b2 dataset.

The proposed Ensemble Concept Mining (ECM) method borrowed ideas from other research papers and added a novel vector based automatic domain relevancy filtering. This expands the ideas from (Pudota, et al. 2010), (Bleik et al., 2010), (Patrick, 2009) and others. Other phrase ranking algorithms work on documents, they had been adopted for clinical narratives. A semantic phrase ranking method has been proposed, which is a slight variation of the domain relevancy filtering. Empirical results showed that our semantic phrase ranking method performs better than the RAKE (Rose et al., 2010) phrase scoring method on the semeval2010 phrase extraction dataset.

The concept extraction activities as described above and in Chapter 4, address the following research question.

Question 1: *How can NLP technology be used to extract concepts from clinical comments to represent a GRiST node or a patient?*

We can use existing NLP technology to extract phrases, but they need to be filtered out by a filtering algorithm to achieve a better outcome. The overall detailed analysis,

comparison of different existing methods and the proposed two stage method is a contribution to the knowledge. We have shown that rather than building supervised models we can combine generic phrase extraction and domain relevancy filtering methods to achieve a similar result.

9.2.2 Semantic Processing

The semantic processing activities as described in Chapter 5 address the following research questions.

Question 2: *How can phrases be stemmed by semantic similarity and how does it compare with string-based similarity?*

The number of extracted phrases can be huge. It is desirable to reduce the number of phrases. For a single word we can apply stemming. For exploration purposes I have compared the string similarity and semantic similarity to reduce the number of phrases. Out of many string similarity algorithms, the Levenshtein distance method looked to be the most promising. Filtering the phrases and only retaining the phrases where similarity was more than a certain value (e.g. 0.80) reduced the number of phrases by up to 2/3 of the original number. Semantic similarity can even work when the string consists of different characters. This was an exploratory work to expand our understanding and it may help other researchers in the future.

Question 3: *Can the semantic vector representation of GRiST nodes help us to find any patterns that may assist us to improve the overall GRiST system?*

Based on the frequently occurring phrases in a specific node and their word vector we have calculated vector representation of each GRiST node. We can see semantically similar nodes appear closer in this analysis. This technique can be used to find semantically similar nodes in an ontology. This could help to review the ontology structure.

Representing a patient by using bag of concepts and other methods has also been discussed. The number of concepts can vary and keeping a fixed dimension of the data for machine learning purposes is desirable. After careful exploration we concluded that using a document vector to represent a patient is far more manageable.

Question 4: *How does the data in the GRiST and its ontological structure relates to other ontology like SNOMED-CT and the implication of these relations on suicide risk?*

The semantic similarity between GRiST nodes and SNOMED-CT concepts were also analysed. We have used word vector based semantic similarity as well as phrase matching to find inter connections between the two ontologies. We have found that this technique finds semantically similar GRiST and Snomed nodes. Distribution of SNOMED-CT concepts across different risk levels has been discussed.

The exploratory works were carried out to improve our understanding about GRiST ontology and NLP technology. This information can be used to review the GRiST structure in the future. This may help future researchers to make a more informed decision.

9.2.3 Risk Prediction

The risk prediction activities as described in Chapter 6 address the following research question.

Question 6: *How do risk predictions produced by using raw text, extracted phrases, word vectors and numerical data compare with each other?*

After concept phase extraction and filtering the next logical step was to use them for risk prediction. This appears to be a very challenging task, especially with the dataset we have. To confirm the results many of the state of the art tools were used. A total of four types of methods were used. The first method used extracted phrases as attributes, the

second method used the full text directly, the third method used document vectors and the final method used numerical data.

A detailed description of the findings of risk prediction is provided in chapter 6. It was found that predicting risk consisting of 10 categories was very challenging. All of the methods produced similar results.

When the number of categories was reduced to three (low, medium and high) accuracy improved significantly. In fact, accuracy can be measured up to 80%. I am aware of the problem of using simple accuracy as a measure of success. When risk was reduced to three categories then most of the risk falls into the low category hence the accuracy was improved. GRiST is mainly designed to take input in a numerical format and inputting comments is optional. Clinicians have not inputted comments in all the required places, which resulted in the missing comments. To address this issue, we have also tried using the association rule mining and information gain method.

The findings substantiate previous findings in the literature (Thompson et al., 2014), (O'Dea et al., 2015). The text analysis may be used to predict risky and non risky patients but to predict a potentially high and low risk category from only text remains to be very challenging. For regression analysis, a dynamic feature selection technique is proposed, which produced better results across the board first time with the GRiST data.

The concept of soft and hard changing symptoms was explored with the GRiST data. For example, if clinicians can intervene in two differing areas, then the system can calculate which action could yield better results in relation to the risk reduction. This technique could allow the GRiST system to provide risk management suggestions to the clinicians in real time.

9.2.4 Association Rule Mining

The node association analysis and activities as described in Chapter 7 addresses the following research question.

Question 6: *How can statistical measures such as chi-square or fp-growth be used to find relationships between the GRiST nodes and how the presence of these relationships might affect risk judgement?*

It is very difficult to mimic the human decision-making process. People can intuitively prioritise things and can take a decision, which a computer program cannot compute. While past suicide attempts may increase the risk but at the same time the presence of regret might mitigate that risk to some extent. This research has tried to find these types of inter-node relationships that may affect risk judgement. To do it mathematically the chi-square, and FP-growth analysis was applied. Our experiments corroborate previous results (Ambert & Cohen, 2012), (Jakulin, 2005) that the attribute interaction may help in classification tasks.

Firstly, the relation between the GRiST nodes have been measured by chi-square and if the p-value was <0.05 then they were assumed to have some significant relationship. There were many nodes that relate to each other in terms of their appearance in the same assessment. The critical question was whether their joint occurrence influenced the overall risk judgement or not. It has been found that in some cases these node relationships do affect the overall risk judgement.

The analysis shows that the node relationships exist, and they appear more in the high-risk category patients than in the lower risk category patients. Considering the implication of this finding, I have looked at node relationships from various angles. The analysis was done first with extracted phrases, then with numerical data. All of these experiments have shown a similar tendency. Presence of the relationships indicated higher risk. This finding is a valuable contribution to the GRiST project.

A novel method has been proposed that can predict suicide risk based on the presence of the node relationships. Empirical results show that while the proposed method was not always able to predict due to no matching rules being found, but when it did the precision of prediction can be up to 90% accurate. This method is very flexible, and we can predict with different levels of confidence. It can predict risk before the assessment is fully completed, hence it is possible to generate a real time alert and the predictive accuracy is better than using regression analysis. This is a significant contribution to the GRiST project.

9.2.5 Reliability of Risk Judgement

The information gain experiments described in Chapter 8 addresses the following research question.

Question 7: *Could the difference between a clinician given and calculated risk be explained by identifying patterns in the raw data particularly by using information theory?*

Information gain is the measure of how much information we know or in other words, how much uncertainty is reduced when we know specific information about an event. It was hypothesised that where there was less information gain by an assessment, the risk difference between the clinicians and consensus risk would be higher. If clinicians did not ask all the relevant questions, then their prediction might not be close to the consensus. We have considered regression generated risk from the numerical data as a consensus risk.

Calculating total information gain is challenging due to the interaction between nodes. We have used the sum of questions weighted by their respective information gain as a relative total information gain of the assessment. The empirical results demonstrate that the risk difference and the total information gain are inversely correlated. This is more so in the case of high-risk category patients. In low risk category patients (risk prediction<5) the clinicians' given and the calculated risk are closely related. There is still a negative correlation, but the impact is much less. However, in the high-risk category patients, the risk difference and total information gain are negatively correlated.

"Relative weights" is a way to partition an MLR model R-square across predictors. They are computed by first transforming p predictors into a new set of p variables that are uncorrelated with one another, yet are correlated as highly as possible with the original predictors (Nimon & Oswald, 2013). We have obtained better results with relative weights. Our results complement the use of information gain for classification by Ambert & Cohen (2012) and the use of explained variance for survival analysis by Maucort-Boulch, Roy, & Stare (2014).

The proposed techniques provide mathematical tools to look at how much information a clinician has taken into account before making a decision. In cases where there is a lack of information, we can dynamically prompt the clinicians to ask high gain questions. This also explains the cases where the clinicians given risk and calculated risk differ significantly. This technique is a valuable contribution to the GRiST system and to other similar systems.

9.3 The Practical Implications

This research has many practical applications in CDSS design and application. In particular, the contribution of this research can be used to improve the interactivity and the reliability of GRiST and similar CDSS systems in general. The following are some of the possible practical applications of this research.

1. We have compared many state of the art phrase extraction algorithms. We have shown that using a two-stage phrase extraction method, which include semantic filtering works better than the manually trained or statistical method. The phrase mining method firstly extracts phrases by using linguistic patterns and then filters them using cosine distance from a domain relevant word list. The domain relevancy is measured by how closely an extracted phrase is related to the main concepts of the domain. It may eliminate the need for human annotation. The practical application of this method has been demonstrated with GRiST, I2B2 and semeval2010 dataset.

2. The concept of semantic stemming is explored to aid in the reduction of the number of extracted phrases. Combining with other filtering this method could be used in a concept mining task. It has been shown that the semantic phrase scoring work better than statistical scoring as found in the RAKE algorithm. A C-language based web service was developed to facilitate the rapid generation of the vector value of a word. This could also help future researchers.

3. The various node relationship finding methods, which have been described in the context of GRiST can also be used in other systems. Knowing how the nodes interact and affect the risk judgement is vital for knowledge engineering. Particularly in the case of GRiST it has been shown that the node relations do affect the risk score. The FP-growth algorithm produced promising results and it can be used in hierarchical data as well. This finding has a direct application in GRiST for alerting the clinician about a potential high suicide risk situation.

4. A simple dynamic feature selection method has been described that can improve the calculation of risk, especially where there are missing data in attributes. It is a very simple technique that can be used in real-time application. Using this technique for the first time, we have been able to calculate suicide risk with better accuracy from the GRiST data.

5. We have proposed a method to determine which intervention measure may be more effective to reduce suicide risk. It may assist in risk management. The proposed method considers both the impact and amenability of the symptoms to adopt better risk management strategy.

6. Using association rule mining to predict suicide risk shows promising results. The predictive accuracy can be up to 90%. In fact, this is the only method that we have found to predict the high risk category accurately. The method is very flexible, and we can predict with various levels of confidence. The rules that have high confidence produce results that are more accurate. This technique can be applied to the GRiST and other similar CDSS systems.

7. A method has been described to explain the differences between clinical and calculated risk by using information gain and relative weights. This can improve the interactivity and reliability of the risk assessment. It may explain why there are differences between the calculated and the clinicians given risk. This adds new capability to the GRiST system for it to become more interactive. It can also suggest which questions can maximise the accuracy of assessments.

We have started with the aim to find various methods to make the GRiST system more interactive and validate its outcome. We have achieved our goals by developing multiple methods to improve the GRiST CDSS. The following figure shows the old and the new proposed workflow of the GRiST system.

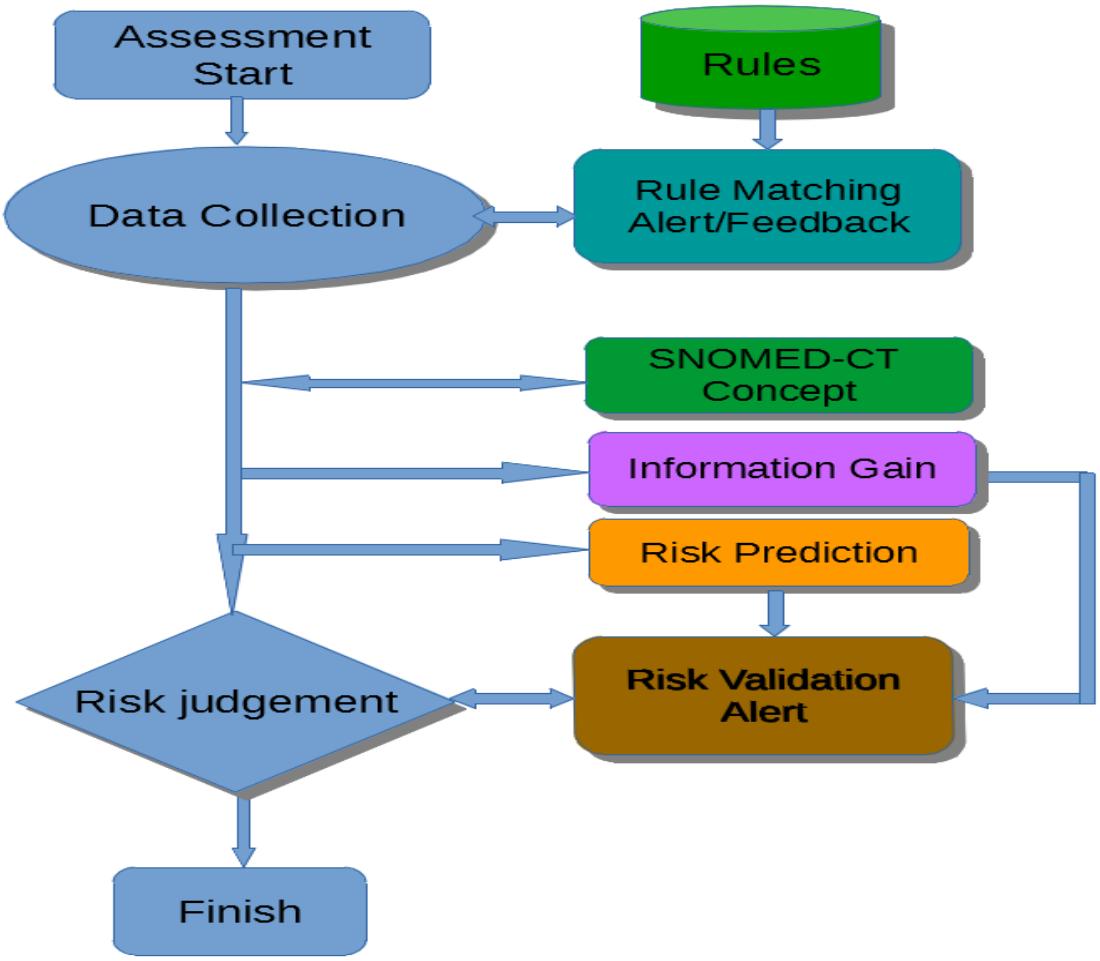

Figure 29 the proposed new GRiST CDSS workflow

All the proposed techniques are generic, and they can be easily adapted for other similar CDSS systems. The application of all the proposed methods may improve the accuracy of the risk judgement and make the system more interactive. This may help gain widespread acceptance of the GRiST system by reducing the complexity of the system and improving the decision accuracy as they are the major factors for a successful CDSS (Al-gamdi, 2014).

9.4 Current Limitations and Future Works

This research has some limitations, which will be discussed along with the proposed future enhancements.

Firstly, the GRiST system was originally designed to collect numerical data. Inputting comments is optional and often they are missing. This could be easily rectified by making some data input mandatory.

Secondly, there is no second judgement available for the same assessment by different clinicians. It could be valuable to use domain experts to re-evaluate some of the assessments and analyse the results.

Thirdly, in GRiST, a filter question may be answered ‘No’ and the nodes underneath would not have any data. In this research, we have only used the ‘scale’ data type, which includes 141 nodes. More research needs to be conducted to explore all other node types.

Finally, this research shows us that an initial assessment and subsequent assessments are different in nature. Some nodes are more amenable by clinical intervention and some are not. Any treatment plans should work with the amenable nodes and those that improve the treatment outcome. As the GRiST data does not include any treatment related information at the moment hence further work needs to be done to bring risk management into the picture.

9.5 Conclusion

The GRiST clinical decision support system is being used to manage mental health related risks. This research uncovered many patterns, described methods and demonstrated techniques that can help to make this system more interactive, improve

performance and ultimately provide better patient care and safety. The findings are not only specific to the GRIST system; they can be adapted for any similar system.

The proposed node relationship analysis could potentially flag high risk patients. The information gain method could evaluate an assessment's reliability and interactively suggest improvements. Repeat assessment and soft, hard node analysis could provide guidance on risk management. Proper implementation of all these measures and methods could make the GRIST system a more interactive and reliable CDSS, which was the main aim of this research.

The actual process of this research activity and having done a lot of reading has enhanced my critical thinking and domain knowledge. It made me more familiar with the related tools and theories. It especially helped me to gain a better understanding of natural language processing and made me more familiar with different machine learning techniques. I hope to continue working in this area of research.

According to the World Health Organisation (WHO) over 800,000 people die due to suicide every year and there are many more who attempt it. I sincerely hope that the use of CDSS such as the GRiST system and the contributions from research like this could save lives in the future.

10 References

Abdel-moneim, W. T., Abdel-Aziz, M. H., & Hassan, M. M. (2013). Clinical Relationships Extraction Techniques from Patient Narratives. *International Journal of Computer Science Issues*, *10*(1), 15. Retrieved from http://arxiv.org/abs/1306.5170

Abraham, S., & Joseph, S. (2016). Rare and frequent weighted itemset optimization using homologous transactions: A rule mining approach. *2015 International Conference on Control, Communication and Computing India, ICCC 2015*, (November), 600–605. https://doi.org/10.1109/ICCC.2015.7432967

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Pas, M., & Soroa, A. (2009). A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches. *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the ACL*, (June), 19–27. https://doi.org/10.3115/1620754.1620758

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, *22*(May), 207–216. https://doi.org/10.1145/170036.170072

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *Proc. 20th IVLDB Conference*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.40.7506&rep=rep1&type=pdf

Ahmad, K., Gillam, L., & Tostevin, L. (1999). Weirdness Indexing for Logical Document Extrapolation and Retrieval. *Trec 8*, 1–8. Retrieved from http://www.google.com/search?sourceid=chrome&ie=UTF-8&q=Weirdness+indexing+for+logical+document+extrapolation+and+retrieval%5Cnpapers2://publication/uuid/4ACADECE-22AF-43B4-9965-093236F0A53F

Ahmed, A. (2011). *Knowledge engineering for mental-health risk assessment and decision support*. Aston University.

Al-gamdi, A. A. (2014). Clinical Decision Support System in HealthCare Industry Success and Risk Factors. *International Journal of Computer Trends and Technology (IJCTT)*, *11*(4), 188–192.

Aleksovska-Stojkovska, L., & Loskovska, S. (2010). Review of Reasoning Methods in Clinical Decision Support Systems. *18th Telecommunications Forum TELFOR*,

1105–1108. Retrieved from http://2010.telfor.rs/files/radovi/TELFOR2010_10_11.pdf

Ali, A., Shamsuddin, S. M., & Ralescu, A. L. (2015). Classification with class imbalance problem: A review. *International Journal of Advances in Soft Computing and Its Applications*, *7*(3), 176–204.

Ambert, K. H., & Cohen, A. M. (2012). κ-information gain scaled nearest neighbors: A novel approach to classifying protein-protein interaction-related documents. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *9*(1), 305–310. https://doi.org/10.1109/TCBB.2011.32

Amro. (2009). What is entropy and information gain? Retrieved January 1, 2015, from http://stackoverflow.com/questions/1859554/what-is-entropy-and-information-gain

Angeli, G., Johnson Premkumar, M. J., & Manning, C. D. (2015). Leveraging Linguistic Structure For Open Domain Information Extraction. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (1), 344–354. https://doi.org/10.3115/v1/P15-1034

Aronson, A. R. (2006). Metamap: Mapping text to the umls metathesaurus. *Bethesda MD NLM NIH DHHS*, 1–26. Retrieved from http://0-skr.nlm.nih.gov.library.law.suffolk.edu/papers/references/metamap06.pdf

Aronson, A. R., & Lang, F.-M. (2010). An overview of MetaMap: historical perspective and recent advances. *Journal of the American Medical Informatics Association*, *17*(3), 229–236. https://doi.org/10.1136/jamia.2009.002733

Bahety, A. (2014). Extension and Evaluation of ID3–Decision Tree Algorithm. *Entropy (S)*, *2*(1), 1–8. Retrieved from http://ssltest.cs.umd.edu/Grad/scholarlypapers/papers/Bahety.pdf

Baker, C. F., Fillmore, C. J., & Lowe, J. B. (1998). The Berkeley FrameNet Project. In *Proceedings of the 17th international conference on Computational linguistics -* (Vol. 1, p. 86). Morristown, NJ, USA: Association for Computational Linguistics. https://doi.org/10.3115/980451.980860

Banfield, R. E., Hall, L. O., Bowyer, K. W., & Kegelmeyer, W. P. (2007). A comparison of decision tree ensemble creation techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(1), 173–180. https://doi.org/10.1109/TPAMI.2007.250609

Bar-Hillel, M. (1980). The base-rate fallacy in probability judgments. *Acta Psychologica*, *44*(3), 211–233. https://doi.org/10.1016/0001-6918(80)90046-3

Batool, R., Khattak, A. M., Kim, T., & Lee, S. (2013). Automatic Extraction and Mapping of Discharge Summary ' s Concepts into SNOMED CT. In *Proceeding of IEEE Eng Med Biol Soc* (pp. 4195–4198). https://doi.org/4195-8

Beeler, P., Bates, D., & Hug, B. (2014). Clinical decision support systems. *Swiss Medical Weekly*, *144*(December), w14073. https://doi.org/10.4414/smw.2014.14073

Bekkerman, R., & Allan, J. (2003). Using Bigrams in Text Categorization. *Work*, *1003*, 1–10. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.1999&rep=rep1&type=pdf

Berner, E. S., & La Lande, T. J. (2007). Overview of Clinical Decision Support Systems. In E. S. Berner (Ed.), *Clinical Decision Support Systems* (pp. 3–22). New York, NY: Springer New York. https://doi.org/10.1007/978-0-387-38319-4_1

Bhatt, U., & Patel, P. (2015). A Novel Approach for Finding Rare Items Based on Multiple Minimum Support Framework. *Procedia Computer Science*, *57*, 1088–1095. https://doi.org/10.1016/j.procs.2015.07.391

Bleik, S., Xiong, W., Wang, Y., & Song, M. (2010). Biomedical concept extraction using concept graphs and ontology-based mapping. *2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 553–556. https://doi.org/10.1109/BIBM.2010.5706627

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching Word Vectors with Subword Information. *arXiv:1607.04606v1 [cs.CL]*. Retrieved from http://arxiv.org/abs/1607.04606

Borgelt, C. (2012). Frequent item set mining. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *2*(6), 437–456. https://doi.org/10.1002/widm.1074

Bouasker, S., & Ben Yahia, S. (2015). Key correlation mining by simultaneous monotone and anti-monotone constraints checking. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing - SAC '15* (pp. 851–856). New York, New York, USA: ACM Press. https://doi.org/10.1145/2695664.2695802

Bouch, J. (2005). Suicide risk: structured professional judgement. *Advances in Psychiatric Treatment*, *11*(2), 84–91. https://doi.org/10.1192/apt.11.2.84

Bridges, S. (2014). Mental health problems. *Health Survey for England 2014*, *1*, 1–16. Retrieved from http://healthsurvey.hscic.gov.uk/support-guidance/public-health/health-survey-for-england-2014/introduction.aspx

Bright, T. J., Wong, A., Dhurjati, R., Bristow, E., Bastian, L., Coeytaux, R. R., … Lobach,

D. (2012). Effect of clinical decision-support systems: a systematic review. *Annals of Internal Medicine*, *157*(1), 29–43. https://doi.org/10.7326/0003-4819-157-1-201207030-00450

Brown, M. (2012). Mental health clinical decision support. Retrieved March 10, 2018, from http://www.clinfowiki.org/wiki/index.php/Mental_health_clinical_decision_support

Buckingham, C. D. (2002). Psychological cue use and implications for a clinical decision support system. *Medical Informatics and the Internet in Medicine*, *27*(4), 237–251. https://doi.org/10.1080/1463923031000063342

Buckingham, C. D., & Adams, A. E. (2008). Cues and knowledge structures used by mental-health professionals when making risk assessments. *Of Mental Health*. Retrieved from http://informahealthcare.com/doi/abs/10.1080/09638230701498374

Buckingham, C. D., & Adams, A. E. (2011). The grist web-based decision support system for mental-health risk assessment and management. Retrieved April 22, 2012, from http://www.egrist.org/sites/egrist.org/files/grist-ita-workshop-2011.pdf

Buckingham, C. D., Ahmed, A., & Adams, A. E. (2007). Using XML and XSLT for flexible elicitation of mental-health risk knowledge. *Medical Informatics and the Internet in Medicine*, *32*(1), 65–81. https://doi.org/10.1080/14639230601097895

Buckingham, C. D., Kearns, G., & Brockie, S. (2004). Developing a computer decision support system for mental health risk screening and assessment. *Current Perspectives*.

Bussone, A., Stumpf, S., & O'Sullivan, D. (2015). The role of explanations on trust and reliance in clinical decision support systems. *Proceedings - 2015 IEEE International Conference on Healthcare Informatics, ICHI 2015*, 160–169. https://doi.org/10.1109/ICHI.2015.26

Byeon, H. (2017). Chi-Square Automatic Interaction Detection Modeling for Predicting Depression in Multicultural Female Students, *8*(12), 179–183.

Cardillo, E. (2015). *Mapping between international medical terminologies to SHN Work Package 3*. Cosenza, Italy.

Carter, T. (2007). *An introduction to information theory and entropy*. *Complex Systems Summer School, Santa Fe*. Santa Fe. Retrieved from http://astarte.csustan.edu/~tom/SFI-CSSS/info-theory/info-lec.pdf

Celikyilmaz, A., Hakkani-Tur, D., & Feng, J. (2010). Probabilistic model-based sentiment analysis of twitter messages. In *2010 IEEE Spoken Language Technology Workshop* (pp. 79–84). IEEE. https://doi.org/10.1109/SLT.2010.5700826

Chao, Y.-C. E., Zhao, Y., Kupper, L. L., & Nylander-French, L. A. (2008). Quantifying the Relative Importance of Predictors in Multiple Linear Regression Analyses for Public Health Studies. *Journal of Occupational and Environmental Hygiene*, *5*(8), 519–529. https://doi.org/10.1080/15459620802225481

Chen, H., Fuller, S. S., Friedman, C., & Hersh, W. (2005). *Medical informatics: knowledge management and data mining in biomedicine*. https://doi.org/10.1007/b135955

Chen, J., Yan, J., Zhang, B., Yang, Q., & Chen, Z. (2006). Diverse Topic Phrase Extraction through Latent Semantic Analysis. In *Sixth International Conference on Data Mining (ICDM'06)* (pp. 834–838). IEEE. https://doi.org/10.1109/ICDM.2006.61

Childs, K., Frick, P. J., Ryals, J. S., Lingonblad, A., & Villio, M. J. (2014). A Comparison of Empirically Based and Structured Professional Judgment Estimation of Risk Using the Structured Assessment of Violence Risk in Youth. *Youth Violence and Juvenile Justice*, *12*(1), 40–57. https://doi.org/10.1177/1541204013480368

Church, K., & Gale, W. (1999). Inverse document frequency (idf): A measure of deviations from poisson. *Natural Language Processing Using Very Large Corpora*, 121–130. https://doi.org/10.1007/978-94-017-2390-9_18

Cobb, R., Puri, S., Wang, D., Cise, D., & Edu, U. F. L. (2013). Knowledge Extraction and Outcome Prediction using Medical Notes.

Cover, T. M., & Thomas, J. A. (2005). Entropy, Relative Entropy, and Mutual Information. In *Elements of Information Theory* (pp. 13–55). Hoboken, NJ, USA: John Wiley & Sons, Inc. https://doi.org/10.1002/047174882X.ch2

Cover, T. M., & Thomas, J. A. (2012). *Elements of Information Theory*. Wiley. Retrieved from https://books.google.co.uk/books?id=VWq5GG6ycxMC

Curtain, C., & Peterson, G. M. (2014). Review of computerized clinical decision support in community pharmacy. *Journal of Clinical Pharmacy and Therapeutics*, *39*(4), 343–348. https://doi.org/10.1111/jcpt.12168

Czibula, G., Czibula, I. G., Cojocar, G. S., & Guran, A. M. (2008). IMASC - An Intelligent MultiAgent System for Clinical Decision Support. In *2008 First International Conference on Complexity and Intelligence of the Artificial and Natural Complex Systems. Medical Applications of the Complex Systems. Biomedical Computing* (pp. 185–190). IEEE. https://doi.org/10.1109/CANS.2008.28

Davies, S. (2013). Annual Report of the Chief Medical Officer 2013, Public Health Priorities: Investing in the evidence, 320. Retrieved from https://www.gov.uk/government/organisations/department-of-health

DE Hert, M., Correll, C. U., Bobes, J., Cetkovich-Bakmas, M., Cohen, D., Asai, I., … Leucht, S. (2011). Physical illness in patients with severe mental disorders. I. Prevalence, impact of medications and disparities in health care. *World Psychiatry : Official Journal of the World Psychiatric Association (WPA)*, *10*(1), 52–77. https://doi.org/10.1002/j.2051-5545.2011.tb00014.x

Department of Health. (2007). Best Practice in Managing Risk. *Best Practice in Risk Management*, (June), 68.

Dharani, M., Menaka, T., & Vinodhini, G. (2014). A Support Vector Machine and Information Gain based Classification Framework for Diabetic Retinopathy Images. *International Journal of Computer Trends and Technology*, *8*(2), 65–69.

Du, W., Du, W., Zhan, Z., & Zhan, Z. (2002). Building decision tree classifier on private data. *Proceedings of the IEEE International Conference on Privacy, Security and Data Mining-Volume 14*, 1–8. Retrieved from http://portal.acm.org/citation.cfm?id=850784

Ducatel, G., Cui, Z., & Azvine, B. (2006). Hybrid ontology and keyword matching indexing system. *Proc. of IntraWebs Workshop at WWW2006*. Retrieved from http://www.ra.ethz.ch/cdstore/www2006/www-sop.inria.fr/acacia/WORKSHOPS/IntraWebs2006/Ducatel_Intrawebs2006.pdf

El-Beltagy, S. R., & Rafea, A. (2009). KP-Miner: A keyphrase extraction system for English and Arabic documents. *Information Systems*, *34*(1), 132–144. https://doi.org/10.1016/j.is.2008.05.002

Elhanan, G., Perl, Y., & Geller, J. (2011). A survey of SNOMED CT direct users, 2010: impressions and preferences regarding content and quality. *Journal of the American Medical Informatics Association*, *18*(Suppl 1), i36–i44. https://doi.org/10.1136/amiajnl-2011-000341

Falzer, P. R. (2013). Valuing Structured Professional Judgment: Predictive Validity, Decision-making, and the Clinical-Actuarial Conflict. *Behavioral Sciences & the Law*, *31*(1), 40–54. https://doi.org/10.1002/bsl.2043

Finkel, J. R., Grenager, T., & Manning, C. D. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. *In Acl*, (1995), 363–370. https://doi.org/10.3115/1219840.1219885

Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in Linguistic Analysis*.

Fournier-Viger, P., Lin, J. C., Gomariz, A., Gueniche, T., Soltani, A., Deng, Z., & Lam, H. T. (2016a). The SPMF Open-Source Data Mining Library Version 2. In B. Berendt,

B. Bringmann, É. Fromont, G. Garriga, P. Miettinen, N. Tatti, & V. Tresp (Eds.) (Vol. 9853, pp. 36–40). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-46131-1_8

Fournier-Viger, P., Lin, J. C., Gomariz, A., Gueniche, T., Soltani, A., Deng, Z., & Lam, H. T. (2016b). The SPMF Open-Source Data Mining Library Version 2. In B. Berendt, B. Bringmann, É. Fromont, G. Garriga, P. Miettinen, N. Tatti, & V. Tresp (Eds.) (Vol. 9853, pp. 36–40). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-46131-1_8

Fournier-Viger, P., Wu, C. W., & Tseng, V. S. (2012). Mining top-K association rules. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *7310 LNAI*, 61–73. https://doi.org/10.1007/978-3-642-30353-1_6

Frantzi, K., Ananiadou, S., & Mima, H. (2000). Automatic recognition of multi-word terms:. the C-value/NC-value method. *International Journal on Digital Libraries*, *3*(2), 115–130. https://doi.org/10.1007/s007999900023

Friedman, C. (2005). Semantic text parsing for patient records. *Medical Informatics*.

Gao, J. B., Zhang, B. W., & Chen, X. H. (2015). A WordNet-based semantic similarity measurement combining edge-counting and information content theory. *Engineering Applications of Artificial Intelligence*, *39*, 80–88. https://doi.org/10.1016/j.engappai.2014.11.009

Gerbier, S., Yarovaya, O., Gicquel, Q., Millet, A.-L., Smaldore, V., Pagliaroli, V., … Metzger, M.-H. (2011). Evaluation of natural language processing from emergency department computerized medical records for intra-hospital syndromic surveillance. *BMC Medical Informatics and Decision Making*, *11*(1), 50. https://doi.org/10.1186/1472-6947-11-50

Giuglea, A., & Moschitti, A. (2004). Knowledge Discovering using FrameNet, VerbNet and PropBank. In *Workshop on Ontology and Knowledge Discovering at ECML'04*. Retrieved from http://olp.dfki.de/pkdd04/giuglea-final.pdf

Gorrell, G., Oduola, S., Roberts, A., Craig, T., Morgan, C., & Stewart, R. (2016). Identifying First Episodes of Psychosis in Psychiatric Patient Records using Machine Learning. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing* (pp. 196–205). Berlin: Association for Computational Linguistics.

Griffith, M. L., Vasilevskis, E. E., Fielstein, E. M., Elkin, P. L., Brown, S. H., Matheny, M. E., … Green, J. K. (2012). Detection of infectious symptoms from VA emergency

department and primary care clinical documentation. *International Journal of Medical Informatics*, *81*(3), 143–156. https://doi.org/10.1016/j.ijmedinf.2011.11.005

Grömping, U. (2006). R package relaimpo: relative importance for linear regression. *Journal Of Statistical Software*, *17*(1), 139–147. https://doi.org/10.1016/j.foreco.2006.08.245

Haldar, R., & Mukhopadhyay, D. (2011). Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach. *arXiv:1101.1232*, (Ld), 286–293. Retrieved from http://arxiv.org/abs/1101.1232

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software. *ACM SIGKDD Explorations Newsletter*, *11*(1), 10. https://doi.org/10.1145/1656274.1656278

Halland, K., Britz, K., & Gerber, A. (2010). Investigations into the use of S NOMED CT to enhance an Open- MRS health information system. *Sacj*, (47), 1–10. https://doi.org/10.18489/sacj.v47i0.14

Hampton, J. A. (2006). Concepts as Prototypes. *Psychology of Learning and Motivation - Advances in Research and Theory*, *46*(2000), 79–113. https://doi.org/10.1016/S0079-7421(06)46003-5

Han, J., Pei, J., Yin, Y., & Mao, R. (2004). Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Mining and Knowledge Discovery*, *8*(1), 53–87. https://doi.org/10.1023/B:DAMI.0000005258.31418.83

Hasan, K. S. K., & Ng, V. (2010). Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art. *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, (August), 365–373. Retrieved from http://dl.acm.org/citation.cfm?id=1944608

Hazlehurst, B., Frost, H. R., Sittig, D. F., & Stevens, V. J. (2005). MediClass: A system for detecting and classifying encounter-based clinical events in any electronic medical record. *Journal of the American Medical Informatics Association : JAMIA*, *12*(5), 517–529. https://doi.org/10.1197/jamia.M1771

Hegazy, S. (2009). A Method for Automatically Eliciting node Weights in a Hierarchical Knowledge-Based Structure for Reasoning with Uncertainty. *International Journal On Advances in Software*, *2*(1), 76–83.

High, R. (2012). The Era of Cognitive Systems: An Inside Look at IBM Watson and How it Works. *International Business Machines Corporation*, *1*(1), 1–14. Retrieved from http://www.redbooks.ibm.com/redpapers/pdfs/redp4955.pdf

Hina, S., Atwell, E., & Johnson, O. (2013). SnoMedTagger: A semantic tagger for

medical narratives. *International Journal of Computational Linguistics and Applications*, *4*(2), 81–99. Retrieved from http://ijcla.bahripublications.com/2013-2/IJCLA-2013-2.pdf#page=81

Hipp, J., Güntzer, U., & Nakhaeizadeh, G. (2000). Algorithms for association rule mining --- a general survey and comparison. *ACM SIGKDD Explorations Newsletter*, *2*(1), 58–64. https://doi.org/10.1145/360402.360421

Hovy, E., Kozareva, Z., & Riloff, E. (2009). Toward completeness in concept extraction and classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2* (pp. 948–957). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dl.acm.org/citation.cfm?id=1699636

Hssina, B., Merbouha, A., Ezzikouri, H., & Erritali, M. (2014). A comparative study of decision tree ID3 and C4.5. *International Journal of Advanced Computer Science and Applications*, (2), 13–19. https://doi.org/10.14569/SpecialIssue.2014.040203

Huang, Y. P., Huang, C. Y., Chen, S. R., Liu, S. I., & Huang, H. C. (2012). Discovering association rules from responded questionnaire for diagnosing geriatric depression. *2012 ICME International Conference on Complex Medical Engineering, CME 2012 Proceedings*, 343–348. https://doi.org/10.1109/ICCME.2012.6275662

IHTSDO. (2014). SNOMED CT Starter Guide. *Snomed*, (July), 1–56. Retrieved from http://www.ihtsdo.org/

Isabel. (2012). Isabel Healthcare | diagnosis tool | medical diagnosis | ddx. Retrieved from http://www.isabelhealthcare.com/home/ourmission

Jakulin, A. (2005). *Machine Learning Based on Attribute Interactions*. *Thesis*. University of Ljubljana. Retrieved from papers2://publication/uuid/E1A2F338-04CB-4A8B-9FB1-A6E209A7C887

Jaspers, M. W. M., Smeulers, M., Vermeulen, H., & Peute, L. W. (2011). Effects of clinical decision-support systems on practitioner performance and patient outcomes: a synthesis of high-quality systematic review findings. *Journal of the American Medical Informatics Association : JAMIA*, *18*(3), 327–334. https://doi.org/10.1136/amiajnl-2011-000094

Jha, A. (2011). The Promise of Electronic Records. *The Journal of the American Medical Association*, *306*(8), 2011–2012.

Jia, P., Zhang, L., Chen, J., Zhao, P., & Zhang, M. (2016). The effects of clinical decision support systems on medication safety: An overview. *PLoS ONE*, *11*(12), 1–17. https://doi.org/10.1371/journal.pone.0167683

Jiang, X., Hu, Y., & Li, H. (2009). A ranking approach to keyphrase extraction. *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '09*, (5), 756. https://doi.org/10.1145/1571941.1572113

Jiang, X., Jao, J., & Neapolitan, R. (2015). Learning predictive interactions using information gain and Bayesian network scoring. *PLoS ONE*, *10*(12), 1–23. https://doi.org/10.1371/journal.pone.0143247

Joos, M., & Zipf, G. K. (1936). The Psycho-Biology of Language. *Language*, *12*(3), 196. https://doi.org/10.2307/408930

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of Tricks for Efficient Text Classification. *arXiv Preprint arXiv:1607.04606*, *2015–Janua*, 4069–4076. Retrieved from http://arxiv.org/abs/1607.01759

Jung, H., Park, H.-A., & Song, T.-M. (2017). Ontology-Based Approach to Social Data Sentiment Analysis: Detection of Adolescent Depression Signals. *Journal of Medical Internet Research*, *19*(7), e259. https://doi.org/10.2196/jmir.7452

Justeson, J. S., & Katz, S. M. (1995). Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, *1*(1), 9–27. https://doi.org/10.1017/S1351324900000048

Kantardzic, M. (2011). Decision Trees and Decision Rules. In *Data Mining* (pp. 169–198). Hoboken, NJ, USA: John Wiley & Sons, Inc. https://doi.org/10.1002/9781118029145.ch6

Kawamoto, K., Del Fiol, G., Lobach, D. F., & Jenders, R. a. (2010). Standards for scalable clinical decision support: need, current and emerging standards, gaps, and proposal for progress. *The Open Medical Informatics Journal*, *4*(919), 235–244. https://doi.org/10.2174/1874431101004010235

Kawamoto, K., Houlihan, C. a, Balas, E. A., & Lobach, D. F. (2005). Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. *BMJ (Clinical Research Ed.)*, *330*(7494), 765. https://doi.org/10.1136/bmj.38398.500764.8F

Kaza, S., & Chen, H. (2008). Evaluating ontology mapping techniques: An experiment in public safety information sharing. *Decision Support Systems*, *45*(4), 714–728. https://doi.org/10.1016/j.dss.2007.12.007

Kiela, D., & Clark, S. (2013). Detecting Compositionality of Multi-Word Expressions using Nearest Neighbours in Vector Space Models. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, (October),

1427–1432.

Kim, J., Chae, Y. M., Kim, S., Ho, S. H., Kim, H. H., & Park, C. B. (2012). A Study on User Satisfaction regarding the Clinical Decision Support System (CDSS) for Medication. *Healthcare Informatics Research*, *18*(1), 35–43. https://doi.org/10.4258/hir.2012.18.1.35

Kiruthika, B., & Roopa, S. N. (2015). A Study on Techniques for Extracting Rare Itemsets. *International Journal of Science and Research*, *4*(6), 2013–2015.

Koeling, R., Tate, A. R., & Carroll, J. A. (2011). Automatically estimating the incidence of symptoms recorded in GP free text notes. In *Proceedings of the first international workshop on Managing interoperability and complexity in health systems - MIXHS '11* (p. 43). New York, New York, USA: ACM Press. https://doi.org/10.1145/2064747.2064757

Kumar, K. P. (2013). Association Rule Mining and Medical Application : A Detailed Survey, *80*(17), 10–19.

Kumar, N., & Srinathan, K. (2008). Automatic keyphrase extraction from scientific documents using N-gram filtration technique. *… of the Eighth ACM Symposium on Document …*, *Sao Paulo,* 199. https://doi.org/10.1145/1410140.1410180

Lacković, I., de Carvalho, P., Zhang, Y. T., & Magjarević, R. (2014). The International Conference on Health Informatics: ICHI 2013, Vilamoura, Portugal on 7-9 November, 2013. *IFMBE Proceedings*, *42*(404), 114–117. https://doi.org/10.1007/978-3-319-03005-0

Lakshmi, K. S., & Kumar, G. S. (2014). Association rule extraction from medical transcripts of diabetic patients. *5th International Conference on the Applications of Digital Information and Web Technologies, ICADIWT 2014*, 201–206. https://doi.org/10.1109/ICADIWT.2014.6814699

Le, Q., & Mikolov, T. (2014). Distributed Representations of Sentences and Documents. *International Conference on Machine Learning - ICML 2014*, *32*, 1188–1196. https://doi.org/10.1145/2740908.2742760

Lee, D. H., Lau, F. Y., & Quan, H. (2010). A method for encoding clinical datasets with SNOMED CT. *BMC Medical Informatics and Decision Making*, *10*, 53. https://doi.org/10.1186/1472-6947-10-53

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*. https://doi.org/citeulike-article-id:311174

Levy, O., & Goldberg, Y. (2014). Dependencybased word embeddings. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, *2*, 302–

308. https://doi.org/10.3115/v1/P14-2050

Litvak, M., Last, M., Aizenman, H., Gobits, I., & Kandel, A. (2011). DegExt — A Language-Independent Graph-Based Keyphrase Extractor. In E. Mugellini, P. Szczepaniak, M. Pettenati, & M. Sokhn (Eds.), *Advances in Intelligent Web Mastering – 3* (Vol. 86, pp. 121–130). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-18029-3_13

Liu, F., Weng, C., & Yu, H. (2012). *Clinical Research Informatics*. (R. L. Richesson & J. E. Andrews, Eds.). London: Springer London. https://doi.org/10.1007/978-1-84882-448-5

Liu, K., Hogan, W. R., & Crowley, R. S. (2011). Natural Language Processing methods and systems for biomedical ontology learning. *Journal of Biomedical Informatics*, *44*(1), 163–179. https://doi.org/10.1016/j.jbi.2010.07.006

Livingston, K. M., Johnson, H. L., Verspoor, K., & Hunter, L. E. (2010). Leveraging Gene Ontology Annotations to Improve a Memory-Based Language Understanding System. *2010 IEEE Fourth International Conference on Semantic Computing*, 40–45. https://doi.org/10.1109/ICSC.2010.62

Longadge, R., Dongre, S. S., & Malik, L. (2013). Class imbalance problem in data mining: review. *International Journal of Computer Science and Network*, *2*(1), 83–87. https://doi.org/10.1109/SIU.2013.6531574

Lopyrev, K. (2014). Learning Distributed Representations of Phrases, 1–5.

Love, T. J., Cai, T., & Karlson, E. W. (2011). Validation of psoriatic arthritis diagnoses in electronic medical records using natural language processing. *Seminars in Arthritis and Rheumatism*, *40*(5), 413–420. https://doi.org/10.1016/j.semarthrit.2010.05.002

Ma, Y., & Distel, F. (2013). Learning Formal Definitions for Snomed CT from Text. In *Artificial Intelligence in Medicine - 14th Conference on Artificial Intelligence in Medicine, {AIME} 2013, Murcia, Spain, May 29 - June 1, 2013. Proceedings* (pp. 73–77). https://doi.org/10.1007/978-3-642-38326-7_11

MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms.ebook2005.pdf*. Copyright Cambridge University Press. https://doi.org/10.1198/jasa.2005.s54

Mahmood, S., Shahbaz, M., & Guergachi, A. (2014). Negative and positive association rules mining from text using frequent and infrequent itemsets. *TheScientificWorldJournal*, *2014*, 973750. https://doi.org/10.1155/2014/973750

Manning, C. D., Bauer, J., Finkel, J., & Bethard, S. J. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *Association for Computational Linguistics*

(ACL) System Demonstrations (pp. 55–60). https://doi.org/10.1.1.650.1022

Manning, C. D., & Klein, D. (2003). Optimization, maxent models, and conditional estimation without magic. *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology Tutorials - NAACL '03*, *5*(June), 8–8. https://doi.org/10.3115/1075168.1075176

Manning, C. D., Raghavan, P., & Schütze, H. (2009). Chapter 8: Evaluation in information retrieval. *Introduction to Information Retrieval*, *10*(c), 1–18. https://doi.org/10.1109/LPT.2009.2020494

Maucort-Boulch, D., Roy, P., & Stare, J. (2014). On a measure of information gain for regression models in survival analysis. *Journal of Applied Statistics*, *41*(12), 2696–2708. https://doi.org/10.1080/02664763.2014.926596

McCallum, A. K. (2002). Mallet: A machine learning for language toolkit. Retrieved from http://mallet.cs.umass.edu

McCart, J. A., Finch, D. K., Jarman, J., Hickling, E., Lind, J. D., Richardson, M. R., … Luther, S. L. (2012). Using ensemble models to classify the sentiment expressed in suicide notes. *Biomedical Informatics Insights*, *5*(Suppl. 1), 77–85. https://doi.org/10.4137/BII.S8931

McDaid, D., Park, A., Iemmi, V., Adelaja, B., Knapp, M., Park, A., & Knapp, M. (2016). Growth in the use of early intervention for psychosis services: An opportunity to promote recovery amid concerns on health care sustainability. *London School Economics*, (January).

McEliece, R. (2013). Entropy and mutual information. In *The Theory of Information and Coding* (pp. 17–49). Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9780511606267.006

Meng, L., Huang, R., & Gu, J. (2013). A Review of Semantic Similarity Measures in WordNet. *International Journal of Hybrid Information Technology*, *6*(1), 1–12.

Mental Health Foundation. (2015). Fundament al Fact s About Mental Health.

Michaels, M. S., Chu, C., Silva, C., Schulman, B. E., & Joiner, T. (2015). Considerations regarding online methods for suicide-related research and suicide risk assessment. *Suicide and Life-Threatening Behavior*, *45*(1), 10–17. https://doi.org/10.1111/sltb.12105

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. *Proceedings of EMNLP*.

Mikolov, T., Corrado, G., Chen, K., & Dean, J. (2013). Efficient Estimation of Word

Representations in Vector Space. *Proceedings of the International Conference on Learning Representations (ICLR 2013)*, 1–12. https://doi.org/10.1162/153244303322533223

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. *Proc. NIPS*, *1*(6), 1–9. https://doi.org/10.1162/jmlr.2003.3.4-5.951

Mikolov, T., Yih, W., & Zweig, G. (2013). Linguistic regularities in continuous space word representations. *Proceedings of NAACL-HLT*, (June), 746–751. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Linguistic+Regularities+in+Continuous+Space+Word+Representations#0%5Cnhttps://www.aclweb.org/anthology/N/N13/N13-1090.pdf

Moore, M., & Loper, K. a. (2011). An Introduction to Clinical Decision Support Systems. *Journal of Electronic Resources in Medical Libraries*, *8*(4), 348–366. https://doi.org/10.1080/15424065.2011.626345

Moral, C., de Antonio, A., Imbert, R., & Ramírez, J. (2014). A survey of stemming algorithms in information retrieval. *Information Research*, *19*(1). https://doi.org/10.9790/0661-17367680

Nagy, S. (2016). *A NOVEL DYNAMIC FEATURE SELECTION AND PREDICTION ALGORITHM FOR CLINICAL DECISIONS INVOLVING HIGH- DIMENSIONAL AND VARIED PATIENT DATA*. Aston University.

National Library of Medicine. (2009). UMLS® Reference Manual - NCBI Bookshelf, (Md). Retrieved from http://www.ncbi.nlm.nih.gov/books/NBK9676/

Naulaerts, S., Meysman, P., Bittremieux, W., Vu, T. N., Berghe, W. Vanden, Goethals, B., & Laukens, K. (2015). A primer to frequent itemset mining for bioinformatics. *Briefings in Bioinformatics*, *16*(2), 216–231. https://doi.org/10.1093/bib/bbt074

Neuman, Y., Cohen, Y., Assaf, D., & Kedma, G. (2012). Proactive screening for depression through metaphorical and automatic text analysis. *Artificial Intelligence in Medicine*, *56*(1), 19–25. https://doi.org/10.1016/j.artmed.2012.06.001

NHS England, the N. C. C. for M. H. and the N. I. for H. and C. E. (2016). Implementing the Early Intervention in Psychosis Access and Waiting Time Standard: Guidance, 57. Retrieved from https://www.england.nhs.uk/mentalhealth/wp-content/uploads/sites/29/2016/04/eip-guidance.pdf

Nimon, K. F., & Oswald, F. L. (2013). Understanding the Results of Multiple Linear Regression: Beyond Standardized Regression Coefficients. *Organizational Research Methods*, *16*(4), 650–674. https://doi.org/10.1177/1094428113493929

O'Dea, B., Wan, S., Batterham, P. J., Calear, A. L., Paris, C., & Christensen, H. (2015). Detecting suicidality on Twitter. *Internet Interventions*, *2*(2), 183–188. https://doi.org/10.1016/j.invent.2015.03.005

Oliva, J., Serrano, J. I., del Castillo, M. D., & Iglesias, Á. (2011). SyMSS: A syntax-based measure for short-text semantic similarity. *Data & Knowledge Engineering*, *70*(4), 390–405. https://doi.org/10.1016/j.datak.2011.01.002

OpenClinical. (2001). Iliad - OpenClinical AI Systems in clinical practice. Retrieved from http://www.openclinical.org/aisp_iliad.html

Orimaye, S. O., Wong, J. S.-M., & Golden, K. J. (2014). Learning Predictive Linguistic Features for Alzheimer's Disease and related Dementias using Verbal Utterances. *Proceedings of the Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, 78–87. Retrieved from http://www.aclweb.org/anthology/W/W14/W14-3210

Osheroff, J. A., Teich, J. M., Middleton, B., Steen, E. B., Wright, A., & Detmer, D. E. (2007). A roadmap for national action on clinical decision support. *JAMIA - Journal of the American Medical Informatics Association*, *14*(2), 141–145. https://doi.org/10.1197/jamia.M2334.Introduction

Padó, S., & Lapata, M. (2007). Dependency-Based Construction of Semantic Space Models. *Computational Linguistics*, *33*(2), 161–199. https://doi.org/10.1162/coli.2007.33.2.161

Pakhomov, S., Buntrock, J., & Duffy, P. (2005). High throughput modularized NLP system for clinical text. *Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions - ACL '05*, 25–28. https://doi.org/10.3115/1225753.1225760

Parameswaran, A., Garcia-Molina, H., & Rajaraman, A. (2010). Towards the web of concepts: extracting concepts from large datasets. *Proc. VLDB Endow.*, *3*(1–2), 566–577.

Park, Y., Byrd, R. J., & Boguraev, B. K. (2002). Automatic glossary extraction. In *Proceedings of the 19th international conference on Computational linguistics -* (Vol. 1, pp. 1–7). Morristown, NJ, USA: Association for Computational Linguistics. https://doi.org/10.3115/1072228.1072370

Patel, R., Jayatilleke, N., Broadbent, M., Chang, C.-K., Foskett, N., Gorrell, G., … Stewart, R. (2015). Negative symptoms in schizophrenia: a study in a large clinical sample of patients using a novel automated method. *BMJ Open*, *5*(9), e007619. https://doi.org/10.1136/bmjopen-2015-007619

Patrick, J. (2009). Intelligent Clinical Notes System: An information retrieval and information extraction system for Clinical Notes. *2009 11th International Conference on E-Health Networking, Applications and Services (Healthcom)*, 108–115. https://doi.org/10.1109/HEALTH.2009.5406206

Patrick, J., Wang, Y., & Budd, P. (2006). Automatic Mapping Clinical Notes to Medical Terminologies. *Australasian Language Technology …*, 75–82. Retrieved from http://www.aclweb.org/anthology/U/U06/U06-1.pdf#page=83

Patrick, J., Wang, Y., & Budd, P. (2007). An automated system for conversion of clinical notes into SNOMED clinical terminology. *Conferences in Research and Practice in Information Technology Series*, *68*, 219–226.

Pawar, P., & Patil, D. (2012). Survey on clinical decision support system. *World Journal of Science and …*, *2*(3), 70–74.

Pedersen, T., & Michelizzi, J. (1998). WordNet :: Similarity - Measuring the Relatedness of Concepts. *HLT-NAACL--Demonstrations '04 Demonstration Papers at HLT-NAACL 2004*, (July), 38–41. https://doi.org/10.3115/1614025.1614037

Pestian, J. P., Matykiewicz, P., Grupp-Phelan, J., Lavanier, S. A., Combs, J., & Kowatch, R. (2008). Using natural language processing to classify suicide notes. *AMIA ... Annual Symposium Proceedings / AMIA Symposium. AMIA Symposium*, (June), 1091.

Petrik, M. L., Gutierrez, P. M., Berlin, J. S., & Saunders, S. M. (2015). Barriers and facilitators of suicide risk assessment in emergency departments: A qualitative study of provider perspectives. *General Hospital Psychiatry*, *37*(6), 581–586. https://doi.org/10.1016/j.genhosppsych.2015.06.018

Pillai, J. (2010). Overview of Itemset Utility Mining and its Applications. *International Journal of Compuer Applications*, *5*(11), 9–13. https://doi.org/10.5120/956-1333

Poria, S., Cambria, E., Winterstein, G., & Huang, G. Bin. (2014). Sentic patterns: Dependency-based rules for concept-level sentiment analysis. *Knowledge-Based Systems*, *69*(1), 45–63. https://doi.org/10.1016/j.knosys.2014.05.005

Porter, M. F. (1980). An algorithm for suffix stripping. *Program: Electronic Library and Information Systems*. https://doi.org/10.1108/eb046814

Poulin, C., Shiner, B., Thompson, P., Vepstas, L., Young-Xu, Y., Goertzel, B., … McAllister, T. (2014). Predicting the risk of suicide by analyzing the text of clinical notes. *PLoS ONE*, *9*(1), 1–7. https://doi.org/10.1371/journal.pone.0085733

Pudota, N., Dattolo, A., Baruzzo, A., Ferrara, F., & Tasso, C. (2010). Automatic keyphrase extraction and ontology mining for content-based tag recommendation.

International Journal of Intelligent Systems, *25*(12), 1158–1186. https://doi.org/10.1002/int.20448

Pudota, N., Dattolo, A., Baruzzo, A., & Tasso, C. (2010). A New Domain Independent Keyphrase Extraction System. *Digital Libraries*.

Pyysalo, S., Ginter, F., Moen, H., Salakoski, T., & Ananiadou, S. (2012). Distributional Semantics Resources for Biomedical Text Processing. *Bio.Nlplab.Org*. Retrieved from http://bio.nlplab.org/pdf/pyysalo13literature.pdf

Rajagopal, D., Cambria, E., Olsher, D., & Kwok, K. (2013). A graph-based approach to commonsense concept extraction and semantic similarity detection. In *Proceedings of the 22nd International Conference on World Wide Web - WWW '13 Companion* (pp. 565–570). New York, New York, USA: ACM Press. https://doi.org/10.1145/2487788.2487995

Ramar, K., & Gurunathan, G. (2016). Technical Review on Ontology Mapping Techniques. *Asian Journal of Information Technology*, *15*(4), 676–688.

Rane, N., & Rao, M. (2013). Association Rule Mining on Type 2 Diabetes using FP-growth association rule. *International Journal Of Engineering And Computer Science*, *2*(8), 4.

Rezaei-yazdi, A. (2015). *Modelling Clinical Expertise for Driving Dynamic Data Collection in a Mental-Health Decision Support System*. Aston University.

Roberts, A., Gaizauskas, R., Hepple, M., & Guo, Y. (2008). Mining clinical relationships from patient narratives. *BMC Bioinformatics*, *9 Suppl 11*, S3. https://doi.org/10.1186/1471-2105-9-S11-S3

Roobaert, D., Karakoulas, G., & Chawla, N. V. (2006). *Information Gain , Correlation and Support Vector Machines*. *Feature Extraction: Foundations and Applications* (Vol. 470).

Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic Keyword Extraction from Individual Documents. *Text Mining: Applications and Theory*, (March), 1–20. https://doi.org/10.1002/9780470689646.ch1

Rouse, M. (2010). What is clinical decision support system (CDSS)? - Definition from WhatIs.com. Retrieved from http://searchhealthit.techtarget.com/definition/clinical-decision-support-system-CDSS

Ryan, A. (2006). Towards semantic interoperability in healthcare: ontology mapping from SNOMED-CT to HL7 version 3. In *Proceedings of the Second Australasian Workshop on Advances in Ontologies* (Vol. 72, pp. 69–74). Retrieved from http://portal.acm.org/citation.cfm?id=1273668

Satpute, V. B. (2014). A Review on Frequent Pattern Mining, *2*(6), 836–841.

Savova, G. K., Masanz, J. J., Ogren, P. V, Zheng, J., Sohn, S., Kipper-Schuler, K. C., & Chute, C. G. (2010). Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association : JAMIA*, *17*(5), 507–513. https://doi.org/10.1136/jamia.2009.001560

Schubert, L. (2015). Semantic Representation. *Proceedings of AAAI*, 4132–4138.

Schuler, K. K. (2005). VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon. *Dissertation Abstracts International, B: Sciences and Engineering*, *66*(6).

Sclano, F., & Velardi, P. (2007). TermExtractor: a Web Application to Learn the Common Terminology of Interest Groups and Research Communities. *Proceedings of the 9th Conference on Terminology and Artificial Intelligence (TIA 2007)*, 8–9.

Shah, A. D., Martinez, C., & Hemingway, H. (2012). The Freetext Matching Algorithm: a computer program to extract diagnoses and causes of death from unstructured text in electronic health records. *BMC Medical Informatics and Decision Making*, *12*(1), 88. https://doi.org/10.1186/1472-6947-12-88

Shahsavarani, A. M., Abadi, E. A. M., Kalkhoran, M. H., Jafari, S., & Qaranli, S. (2015). Clinical Decision Support Systems (CDSSs): State of the art Review of Literature. *International Journal of Medical Reviews*, *2*(4), 299–308. Retrieved from http://journals.bmsu.ac.ir/ijmr/index.php/ijmr/article/view/137

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, *27*(July 1928), 379–423. https://doi.org/10.1145/584091.584093

Sharma, A., Swaminathan, R., & Yang, H. (2010). A Verb-Centric Approach for Relationship Extraction in Biomedical Text. *2010 IEEE Fourth International Conference on Semantic Computing*, 377–385. https://doi.org/10.1109/ICSC.2010.14

Sharma, D., & Cse, M. (2012). Stemming Algorithms: A Comparative Study and their Analysis. *International Journal of Applied Information Systems*, *4*(3), 7–12. Retrieved from http://research.ijais.org/volume4/number3/ijais12-450655.pdf

Sharma, V. K., Krishna, M., Lepping, P., Palanisamy, V., Kallumpuram, S. V., Mottram, P., … Copeland, J. R. M. (2010). Validation and feasibility of the Global Mental Health Assessment Tool--Primary Care Version (GMHAT/PC) in older adults. *Age and Ageing*, *39*(4), 496–499. https://doi.org/10.1093/ageing/afq050

Shi, H., Zhou, G., Qian, P., & Li, X. (2009). Semantic Role Labeling Based on Dependency Tree with Multi-features. *2009 International Joint Conference on*

Bioinformatics, Systems Biology and Intelligent Computing, 584–587. https://doi.org/10.1109/IJCBS.2009.99

Shortliffe, E. H. (1977). Mycin: A Knowledge-Based Computer Program Applied to Infectious Diseases. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, 66–69. Retrieved from http://www.ncbi.nlm.nih.gov.proxygw.wrlc.org/pmc/articles/PMC2464549/

Shortliffe, E. H., & Cimino, J. J. (Eds.). (2014). *Biomedical Informatics* (Vol. 12). London: Springer London. https://doi.org/10.1007/978-1-4471-4474-8

Shtatland, E. S., & Barton, M. B. (1997). INFORMATION AS A UNIFYING MEASURE OF FIT IN SAS® STATISTICAL MODELING PROCEDURES. *Northeast SAS Users Group NESUG'97 Proceedings*, 875--880.

Siddiqi, S., & Sharan, A. (2015). Keyword and Keyphrase Extraction Techniques : A Literature Review. *Nternational Journal of Computer Applications*, *109*(2), 18–23.

Singh, S. P. (2010). Early intervention in psychosis. *The British Journal of Psychiatry : The Journal of Mental Science*, *196*(5), 343–345. https://doi.org/10.1192/bjp.bp.109.075804

Sondhi, P., Sun, J., Tong, H., & Zhai, C. (2012). SympGraph. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12* (p. 1167). New York, New York, USA: ACM Press. https://doi.org/10.1145/2339530.2339712

Stahl, J. E. (2008). Modelling methods for pharmacoeconomics and health technology assessment: an overview and guide. *PharmacoEconomics*, *26*(2), 131–148. https://doi.org/10.1016/j.jval.2014.12.014

Stenzhorn, H., Pacheco, E. J., Nohama, P., & Schulz, S. (2009). Automatic mapping of clinical documentation to SNOMED CT. In *Studies in Health Technology and Informatics* (Vol. 150, pp. 228–232). https://doi.org/10.3233/978-1-60750-044-5-228

Stojkovska, L. A., Loskovska, S., & Member, S. (2010). Clinical decision support systems: Medical knowledge acquisition and representation methods. *2010 IEEE International Conference on Electro Information Technology*, 1–6. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5612183

Subhashini, R., & Kumar, V. J. S. (2010). Shallow NLP techniques for noun phrase extraction. *Trendz in Information Sciences & Computing(TISC2010)*, 73–77. https://doi.org/10.1109/TISC.2010.5714612

Thompson, P., Bryan, C., & Poulin, C. (2014). Predicting military and veteran suicide risk: Cultural aspects. *Proceedings of the Workshop on Computational Linguistics*

and Clinical Psychology: From Linguistic Signal to Clinical Reality, 1–6. Retrieved from http://www.aclweb.org/anthology/W/W14/W14-3201

Tomokiyo, T., & Hurst, M. (2003). A language model approach to keyphrase extraction. *Proceedings of the ACL 2003 Workshop on Multiword Expressions Analysis, Acquisition and Treatment -*, *18*, 33–40. https://doi.org/10.3115/1119282.1119287

Turney, P. D. (2013). Distributional Semantics Beyond Words: Supervised Learning of Analogy and Paraphrase. *Tacl*, *1*, 353–366. Retrieved from http://arxiv.org/abs/1310.5042

Uzuner, Ö., South, B. R., Shen, S., & DuVall, S. L. (2011). 2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, *18*(5), 552–556. https://doi.org/10.1136/amiajnl-2011-000203

Vinh, N. X., Epps, J., & Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, *11*, 2837–2854. https://doi.org/10.1182/blood-2008-03-145946

Wan, X., & Xiao, J. (2008). Single Document Keyphrase Extraction Using Neighborhood Knowledge. *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, 855–860. https://doi.org/10.1145/1740592.1740596

Wang, X., Chused, A., Elhadad, N., Friedman, C., & Markatou, M. (2008). Automated knowledge acquisition from clinical narrative reports. *AMIA ... Annual Symposium Proceedings / AMIA Symposium. AMIA Symposium*, 783–787.

Wang, X., McCallum, A., & Wei, X. (2007). Topical N-grams: Phrase and topic discovery, with an application to information retrieval. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 697–702. https://doi.org/10.1109/ICDM.2007.86

Wang, X., Mu, D., & Fang, J. (2008). Improved Automatic Keyphrase Extraction by Using Semantic Information. *2008 International Conference on Intelligent Computation Technology and Automation (ICICTA)*, 1061–1065. https://doi.org/10.1109/ICICTA.2008.180

Whetzel, P. L., Noy, N., Shah, N., Alexander, P., Dorf, M., Fergerson, R., … Musen, M. (2011). BioPortal: Ontologies and integrated data resources at the click of a mouse. *CEUR Workshop Proceedings*, *833*(May), 292–293. https://doi.org/10.1093/nar/gkp440

Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., & Nevill-Manning, C. G. (1999).

KEA: Practical Automatic Keyphrase Extraction. *In Proceedings of the Fourth ACM Conference on Digital Libraries, ACM*, 254–261. https://doi.org/10.1.1.17.7577

World Health Organization. (2013). Mental Health Action Plan 2013-2020. *WHO Library Cataloguing-in-Publication DataLibrary Cataloguing-in-Publication Data*, 1–44. https://doi.org/ISBN 978 92 4 150602 1

Wu, B., Lu, X., & Duan, H. (2008). An Automatic Knowledge Acquisition Mechanism for Independent Inference Engine Module of CDSS. *2008 2nd International Conference on Bioinformatics and Biomedical Engineering*, 1293–1296. https://doi.org/10.1109/ICBBE.2008.651

Wu, Y., Denny, J. C., Rosenbloom, S. T., Miller, R. A., Giuse, D. A., & Xu, H. (2012). A comparative study of current Clinical Natural Language Processing systems on handling abbreviations in discharge summaries. *AMIA ... Annual Symposium Proceedings / AMIA Symposium. AMIA Symposium*, *2012*, 997–1003. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/23304375%5Cnhttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3540461

Yang, H., Spasic, I., Keane, J. a, & Nenadic, G. (2012). A text mining approach to the prediction of disease status from clinical discharge summaries. *Journal of the American Medical Informatics Association : JAMIA*, *16*(4), 596–600. https://doi.org/10.1197/jamia.M3096

Yang, Yang, Alistair Willis, Anne de Roeck, & Bashar Nuseibeh. (2012). A Hybrid Model for Automatic Emotion Recognition in Suicide Notes. *Biomedical Informatics Insights*, *5*(Suppl. 1), 17. https://doi.org/10.4137/BII.S8948

Yu, H., Ho, C.-H., Juan, Y., & Lin, C. (2013). *Libshorttext: a library for short-text classification and analysis*. Retrieved from http://ntu.csie.org/~cjlin/papers/libshorttext.pdf%5Cnhttp://www.csie.ntu.edu.tw/~cjlin/libshorttext

Zhang, S., & Wu, X. (2011). Fundamentals of association rules in data mining and knowledge discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1*(2), 97–116. https://doi.org/10.1002/widm.10

Zhang, W., Ma, D., & Yao, W. (2014). Medical diagnosis data mining based on improved Apriori algorithm. *Journal of Networks*, *9*(5), 1339–1345. https://doi.org/10.4304/jnw.9.5.1339-1345

Zhang, Z., Gao, J., & Ciravegna, F. (2016). JATE 2.0 : Java Automatic Term Extraction with Apache Solr. *10th International Conference on Language Resources and*

Evaluation (LREC'16), Portorôz, Slovenia, (May), 2262–2269.

Zhang, Z., Iria, J., Brewster, C., & Ciravegna, F. (2008). A comparative evaluation of term recognition algorithms. *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, 2108–2113. https://doi.org/10.1117/12.326718

Zhao, W. X., Jiang, J., He, J., Song, Y., Achananuparp, P., Lim, E.-P., & Li, X. (2011). Topical keyphrase extraction from Twitter. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 379–388. Retrieved from http://dl.acm.org/citation.cfm?id=2002472.2002521

Zhou, D., & He, Y. (2008). Extracting interactions between proteins from the literature. *Journal of Biomedical Informatics*, *41*(2), 393–407. https://doi.org/10.1016/j.jbi.2007.11.008

Zhou, D., Zhong, D., & He, Y. (2014). Event trigger identification for biomedical events extraction using domain knowledge. *Bioinformatics*, *30*(11), 1587–1594. https://doi.org/10.1093/bioinformatics/btu061

Zhou, L., & Yau, S. (2007). Efficient association rule mining among both frequent and infrequent items. *Computers and Mathematics with Applications*, *54*(6), 737–749. https://doi.org/10.1016/j.camwa.2007.02.010

Zhou, X., Han, H., Chankai, I., Prestrud, A., & Brooks, A. (2006). Approaches to text mining for clinical medical records. *Proceedings of the 2006 ACM Symposium on Applied Computing - SAC '06*, 235. https://doi.org/10.1145/1141277.1141330

Zibran, M. F. (2015). Chi-squared test of independence. *Handbook of Biological Statistics*, 1–7. Retrieved from http://www.biostathandbook.com/chiind.html

ZynxHealth. (2012). ZynxEvidence Clinical Evidence | Clinical Content and Guidelines Supporting Evidence-Based Practice | Zynx Health. Retrieved from http://www.zynxhealth.com/Solutions/ZynxEvidence.aspx

Appendix A **GRiST Ontology**

Following text shows a section of the GRiST ontology to demonstrate its logical structure. Question type, node label, etc. all attributes are removed to make the text size small.

<node code="mental-health-risk" >
<node code="suic" >
<node code="suic-specific" >
<node code="suic-past-att" >
<node code="suic-occur" >
<node code="suic-most-rec" />
<node code="suic-patt-att" >
<node code="suic-first-occ" />
<node code="suic-how-many" />
<node code="suic-escalate" />
</node>
</node>
<node code="suic-prep-serious-at" >
<node code="suic-note-prev" />
<node code="suic-ser-method" >
<node code="suic-discovery" />
<node code="suic-lethality" />
</node>
</node>
<node code="suic-person-per" >
<node code="suic-thght-prev" >
<node code="suic-ser-succd" />
<node code="suic-regret" />
</node>
<node code="suic-leth-insght" />
</node>
</node>
<node code="suic-curr-sit-behav" >

<node code="suic-curr-int" >
<node code="suic-plans" >
<node code="suic-plan-real" />
<node code="suic-steps-takn" />
<node code="suic-prosp-leth" />
</node>
<node code="suic-int-inform" />
<node code="suic-eol-prep" />
<node code="suic-s-h-behv" />
<node code="suic-bhvr-const" >
<node code="insight-resp" />
<node code="suic-rel-belief" />
</node>
</node>
<node code="suic-int-p-trig" >
<node code="suic-pot-trig" />
<node code="suic-p-trig-mtch" />
<node code="suic-fam-hist" />
</node>
<node code="suic-ideation" >
<node code="suic-id-control" />
<node code="suic-id-hi-risk" />
<node code="suic-id-freq" />
<node code="suic-id-strngth" />
</node>
<node code="suic-app-behvr" >
<node code="suic-phys-indic" >
<node code="sn-appearnce" />
<node code="gen-sh-cuts" />
</node>
<node code="gen-presentation" />
</node>
</node>
</node>
<node code="gen-direct" />

</node>
</node>

Table 77 Grist ontology node name and description

Grist Node	Description or Meaning
suic	ending your own life
suic_patt_att	pattern of attempts to end your life
suic_prep_serious_at	preparation and seriousness of attempts to end your life
suic_discovery	chance of being found after attempting to end your life
suic_lethality	how dangerous were your attempts to end your life
suic_person_per	what you think now about past attempts to end your life
suic_ser_succd	How much did you want to end your life
suic_regret	regret trying to end your life
suic_leth_insght	awareness of how dangerous were previous attempts to end your life
suic_plans	plans and methods for ending your life
suic_plan_real	realism of plan to end your life
suic_steps_takn	steps taken towards carrying out your plan to end your life
suic_prosp_leth	dangerousness of method to end your life
suic_eol_prep	making end of life preparations
suic_s_h_behv	dangerous self-harming
suic_bhvr_const	things that might help stop you ending your life
suic_int_p_trig	triggers for ending your life
suic_pot_trig	potential triggers for ending your life
suic_p_trig_mtch	match between current triggers and dangerous ones in the past
suic_ideation	thoughts about ending your life
suic_id_control	ability to control thoughts about ending your life
suic_id_hi_risk	very risky thoughts about ending your life
suic_id_strngth	strength
suic_app_behvr	your appearance and the distress it reflects
gen_presentation	your behavioural presentation
sh	Harming yourself

Grist Node	Description or Meaning
sh_patt_of_eps	pattern of self-harming
sh_seriousns_eps	seriousness of self-harming
sh_hlp_after	chance of being helped after self-harming
sh_lethality_mth	dangerousness of self-harming method
sh_for_hlp_diff	self-harming to get help
sh_pot_triggs_p	triggers for self-harm
sh_pot_triggs	potential triggers for self-harm
sh_pot_trigs_mtch	match between current trggers and dangerous ones in the past
sh_ideation	self-harming thoughts
sh_strength	strength
gen_presentation	your appearance and the distress it reflects
hto	harming others or damaging property
hto_emotional_ep	seriousness of emotional harm to others
hto_violent	seriousness of violence or abuse
hto_dest_prprty	seriousness of damage to property
hto_fire_setting	seriousness of fire-setting
hto_to_anmls	seriousness of harming animals
hto_curr_persp_ep	thoughts about harm or damage you caused in the past
hto_intention	intention to cause harm or damage
hto_means_plan	ability to carry out your plan to cause harm or damage
hto_steps_plan	putting your plan for harm or damage into action
app_harm_dam	your appearance and the threat it reflects
hto_constr_bhvr	things that stop you causing harm or damage
pot_trig_hto	triggers for harm or damage
hto_pot_trig	potential triggers of harm or damage
hto_pot_trig_mtch	match between current triggers and dangerous ones in the past
hto_fam_hist	family history of causing harm or damage
hto_ideation_vio	violent thoughts
hto_hi_rsk_ideatn	high risk violent thoughts
hto_strgth_ideatn	strength
hto_ideatn_link	violent thoughts about real people
gen_presentation	your appearance and behaviour
risk_dep	risk to dependents

Grist Node	Description or Meaning
gen_presentation	your appearance and behaviour
sn	not looking after yourself
sn_app_behavr	your appearance and the lack of care for yourself it reflects
sn_hair_clothes	hair and clothes
sn_hygiene	personal hygiene
sn_recnt_app_chnge	recent changes in your appearance
sn_skin	skin
vuln_su	feeling vulnerable
app_vuln_abuse	your appearance and the vulnerability to abuse it reflects
sex_vuln	sexual vulnerability
phys_vuln	physical vulnerability
emot_vuln	emotional vulnerability
finan_vuln	financial vulnerability
dis_conf	confusion and disorientation
carers	depending on carers
gen_liv_skills	life skills
gen_mood_swings	mood swings
gen_negative_self	negative feelings about myself
gen_angry_emotns	feeling angry
gen_anx_emotns	anxiety
gen_helpless	feeling helpless
gen_sad	feeling sad
gen_hopeless	hopefulness
gen_plans_future	plans for the future
gen_life_not_livng	enjoyment of your life
grandiosity	feeling all-important
worthlessness	self-worth
gen_empathy_abil	empathy
gen_coping_abil	coping with major life stresses
gen_decision	making decisions
gen_insght_behvr	understanding risk taking
gen_resp_impct_oth	taking responsibility for the risks I take
gen_nd_hlp_diff	my need for help

Grist Node	Description or Meaning
gen_mania	feeling manic
gen_voice_dang_s	danger of the voices to you
gen_voice_dang_o	danger of the voices to others
gen_prob_act_voice	acting on the voices
gen_paran_del_spec	about specific people
gen_paran_del_pers	feeling under threat
gen_prob_act_par_del	acting on your paranoid thoughts
gen_mentl_insght	understanding my mental-health problems
gen_phys_hlth_prb	my physical health
gen_epi_sieze	fits/epilepsy
gen_com_imp	communication difficulties
gen_phys_hlth_det	worsening physical health
gen_meds_concord	following health professionals' advice
gen_serv_perc_supp	service support
gen_serv_last_acc	regular use of services
gen_med_perc_benft	benefit from treatment
gen_phys_withd	physical withdrawal from the world
gen_motivation	my motivation in life
gen_listless	energy
gen_net_relat	network of people in my life
gen_relat_detr	relationships that are bad for you
gen_relat_detr_chg	changes for the worse in relationships
gen_isol_accom	living in an isolated place
gen_neigbrhd_rsky	my neighbourhood
gen_accom_hm_care	caring for my home
gen_accom_habitbl	home comfort
gen_perc_debt_anx	debt
gen_poverty	cost of living
gen_job_chg_frq	my work history
gen_rec_bad_job_ch	unhelpful changes to employment
gen_rsk_behavr	risky activities
gen_unint_risk_behavr	carelessness
gen_sleep_dist	sleep

Grist Node	Description or Meaning
gen_diet_weigt_chg	weight change
gen_diet_drink	fluid intake
gen_unusl_rec_bhvr	not being myself
gen_chall_bhvr	being challenging
gen_day_actvty	my general daily activity
gen_day_struct	structure in my lfe
gen_alc_misuse	bad effects of alcohol
gen_drug_misuse	bad effects of drugs
gen_env_grew_up	environment I grew up in
gen_educ_expr	school days
gen_rapport	rapport
gen_responsve	responsiveness
gen_gut_assmnt	uneasiness about you
gen_risk_aggrsv	sounding aggressive
gen_risk_upbeat	sounding depressed
gen_coherence	making sense to others
gen_distrss_b_lang	body language and distress
gen_low_mood	body language and low mood
gen_threat_move	body language and aggression
gen_detached	seeming preoccupied or detached
gen_avoid_eye_contact	eye contact
gen_congruence	presenting a coherent story

Appendix B **GRiST Node Relationships**

Table 78 Node to node cosine similarity

Node name	Similar nodes
gen_concentr	gen_concentr=1.0, gen_anx_emotns=0.8, gen_mania=0.8, gen_congruence=0.8, gen_sleep_dist=0.8, gen_mood_swings=0.8, gen_distress=0.8, gen_cog_think_mem=0.8, hto_pot_trig=0.8, gen_unusl_rec_bhvr=0.8, gen_helpless=0.84
gen_plans_future	gen_plans_future=1.0, gen_helpless=0.9, gen_life_not_livng=0.9, gen_dependence=0.9, worthlessness=0.9, gen_net_relat=0.9, gen_sad=0.9, gen_phys_withd=0.9, gen_relat_supp=0.9, suic_p_trig_mtch=0.9, gen_relat_detr_chg=0.92
hto_violent	hto_violent=1.0, hto_dest_prprty=0.9, hto_curr_persp_ep=0.8, app_vuln_abuse=0.8, hto_means_plan=0.8, hto_emotional_ep=0.8, hto=0.8, sex_vuln=0.8, gen_relat_detr=0.7, hto_to_anmls=0.7, risk_dep=0.76
suic_id_control	suic_id_control=1.0, suic_id_hi_risk=0.9, suic_ser_succd=0.9, sh_pot_triggs=0.9, sh=0.9, gen_unusl_rec_bhvr=0.9, suic_p_trig_mtch=0.9, gen_impulse=0.9, suic_planning=0.9, hto_pot_trig=0.9, vuln_su=0.93
suic_eol_prep	suic_eol_prep=1.0, suic_id_strngth=0.8, gen_dependence=0.8, gen_gut_assmnt=0.8, finan_vuln=0.8, suic_id_hi_risk=0.8, gen_life_not_livng=0.8, gen_prob_act_par_del=0.8, gen_net_relat=0.8, sh_planning=0.8, gen_insght_behvr=0.85
sn_hygiene	sn_hygiene=1.0, sn_hair_clothes=0.8, sn_recnt_app_chnge=0.8, sn_skin=0.8, gen_app_diet=0.8, gen_serv_perc_supp=0.8, gen_diet_eating=0.8, gen_accom_hm_care=0.8, gen_liv_skills=0.8, gen_decision=0.7, gen_phys_hlth_det=0.79
gen_rec_bad_job_ch	gen_rec_bad_job_ch=1.0, gen_job_chg_frq=0.9, gen_perc_debt_anx=0.9, gen_poverty=0.9, dis_conf=0.8, gen_net_relat=0.8, gen_relat_detr_chg=0.8, gen_dependence=0.8, worthlessness=0.8, gen_plans_future=0.8, gen_accom_habitbl=0.87
gen_meds_concord	gen_meds_concord=1.0, gen_alc_misuse=0.9, gen_med_perc_benft=0.9, gen_nd_hlp_diff=0.9, gen_drug_misuse=0.9, suic_lethality=0.9, gen_mentl_insght=0.9, sh=0.9, suic_planning=0.9, sn=0.8, suic_leth_insght=0.88
gen_rsk_behavr	gen_rsk_behavr=1.0, gen_unint_risk_behavr=0.9, gen_alc_misuse=0.9, gen_impulse=0.9, gen_insght_behvr=0.9, sex_vuln=0.9, gen_drug_misuse=0.9, gen_neigbrhd_rsky=0.9, suic_planning=0.9, suic_lethality=0.9, suic_id_control=0.89
gen_unint_risk_behavr	gen_unint_risk_behavr=1.0, gen_rsk_behavr=0.9, gen_impulse=0.9, suic_lethality=0.9, gen_alc_misuse=0.9, suic_planning=0.9, suic_leth_insght=0.9, gen_insght_behvr=0.9, suic_id_control=0.9, gen_drug_misuse=0.8, gen_neigbrhd_rsky=0.88

Node name	Similar nodes
gen_phys_hlth_det	gen_phys_hlth_det=1.0, gen_phys_hlth_disa=0.9, gen_phys_hlth_pain=0.9, gen_app_diet=0.8, gen_com_imp=0.8, sn=0.8, gen_cog_think_mem=0.8, sh_lethality_mth=0.8, sn_skin=0.8, gen_diet_eating=0.8, gen_helpless=0.81
Hto	hto=1.0, risk_dep=0.9, phys_vuln=0.9, hto_curr_persp_ep=0.9, gen_relat_detr=0.9, gen_chall_bhvr=0.9, hto_pot_trig=0.9, sex_vuln=0.9, vuln_su=0.8, app_vuln_abuse=0.8, gen_prob_act_par_del=0.89
gen_jealous	gen_jealous=1.0, emot_vuln=0.9, gen_prob_act_par_del=0.9, hto_pot_trig=0.9, gen_chall_bhvr=0.8, phys_vuln=0.8, risk_dep=0.8, gen_angry_emotns=0.8, gen_relat_detr_chg=0.8, gen_paran_del_spec=0.8, gen_unusl_rec_bhvr=0.87
sh_pot_triggs	sh_pot_triggs=1.0, suic_p_trig_mtch=0.9, hto_pot_trig=0.9, sh=0.9, suic_ser_succd=0.9, vuln_su=0.9, suic_id_hi_risk=0.9, gen_unusl_rec_bhvr=0.9, suic_id_control=0.9, suic_pot_trig=0.9, gen_distress=0.94
gen_resp_impct_oth	gen_resp_impct_oth=1.0, gen_insght_behvr=0.9, hto_pot_trig_mtch=0.9, gen_neigbrhd_rsky=0.9, gen_prob_act_par_del=0.9, gen_impulse=0.9, hto_curr_persp_ep=0.9, finan_vuln=0.9, gen_relat_detr=0.9, gen_reliable=0.9, suic_id_hi_risk=0.90
sn	sn=1.0, gen_app_diet=0.9, gen_nd_hlp_diff=0.9, gen_mentl_insght=0.9, gen_helpless=0.9, gen_cog_think_mem=0.9, sh=0.9, gen_diet_eating=0.9, suic_planning=0.9, vuln_su=0.9, gen_unusl_rec_bhvr=0.90
hto_ideatn_link	hto_ideatn_link=1.0, risk_dep=0.9, hto_hi_rsk_ideatn=0.8, gen_prob_act_par_del=0.8, phys_vuln=0.8, hto_curr_persp_ep=0.8, hto=0.8, gen_paran_del_pers=0.8, gen_voice_dang_s=0.8, emot_vuln=0.8, app_vuln_abuse=0.84
gen_cog_think_mem	gen_cog_think_mem=1.0, sn=0.9, gen_helpless=0.9, gen_app_diet=0.9, gen_diet_eating=0.8, gen_sleep_dist=0.8, gen_phys_withd=0.8, gen_life_not_livng=0.8, gen_nd_hlp_diff=0.8, gen_decision=0.8, gen_plans_future=0.87
hto_curr_persp_ep	hto_curr_persp_ep=1.0, risk_dep=0.9, hto=0.9, app_vuln_abuse=0.9, gen_relat_detr=0.9, gen_resp_impct_oth=0.9, gen_prob_act_par_del=0.9, sex_vuln=0.9, phys_vuln=0.8, hto_hi_rsk_ideatn=0.8, gen_neigbrhd_rsky=0.89
gen_app_diet	gen_app_diet=1.0, sn=0.9, gen_diet_eating=0.9, gen_helpless=0.9, gen_nd_hlp_diff=0.9, gen_med_perc_benft=0.9, gen_plans_future=0.9, gen_eating_dis=0.9, gen_mentl_insght=0.9, gen_sleep_dist=0.9, gen_life_not_livng=0.90
gen_paran_del_pers	gen_paran_del_pers=1.0, gen_prob_act_par_del=0.9, hto_curr_persp_ep=0.8, risk_dep=0.8, gen_paran_del_spec=0.8, hto_hi_rsk_ideatn=0.8, gen_resp_impct_oth=0.8, hto_ideatn_link=0.8, gen_gut_assmnt=0.8, phys_vuln=0.8, hto_pot_trig_mtch=0.84
hto_fam_hist	hto_fam_hist=1.0, gen_relat_detr=0.9, sex_vuln=0.9, app_vuln_abuse=0.9, hto=0.8, hto_emotional_ep=0.8, gen_resp_impct_oth=0.8, hto_curr_persp_ep=0.8, risk_dep=0.8, gen_env_grew_up=0.8, gen_relat_supp=0.86

Table 79 Node to node correlation by mg-value

Node name	Related nodes
suic_planning	suic_planning=1.00, suic_discovery=0.63, suic_ser_succd=0.61, sh_planning=0.50, suic_lethality=0.48, suic_id_hi_risk=0.38, suic_id_strngth=0.38, sh_lethality_mth=0.36, suic_eol_prep=0.35, suic_regret=0.34
suic_discovery	suic_discovery=1.00, suic_planning=0.63, suic_ser_succd=0.62, suic_lethality=0.48, sh_planning=0.37, sh_hlp_after=0.36, suic_id_strngth=0.32, suic_eol_prep=0.32, suic_regret=0.31, app_harm_dam=0.29
suic_lethality	suic_lethality=1.00, suic_ser_succd=0.65, suic_planning=0.48, suic_discovery=0.48, sh_lethality_mth=0.40, suic_id_hi_risk=0.38, suic_id_strngth=0.38, suic_id_control=0.34, suic_prosp_leth=0.32, hto_fire_setting=0.27
suic_ser_succd	suic_ser_succd=1.00, suic_lethality=0.65, suic_discovery=0.62, suic_planning=0.61, suic_id_hi_risk=0.44, suic_id_strngth=0.43, suic_id_control=0.39, suic_regret=0.38, sh_planning=0.34, sh_hlp_after=0.32
suic_regret	suic_regret=1.00, suic_id_hi_risk=0.49, suic_id_strngth=0.45, gen_life_not_livng=0.45, suic_id_control=0.44, gen_plans_future=0.41, suic_p_trig_mtch=0.38, suic_ser_succd=0.38, app_harm_dam=0.37, suic_leth_insght=0.37
suic_leth_insght	suic_leth_insght=1.00, suic_regret=0.37, gen_mentl_insght=0.35, app_harm_dam=0.32, suic_id_control=0.32, gen_learn_disab=0.32, gen_insght_behvr=0.31, suic_id_hi_risk=0.30, vuln_su=0.29, sex_vuln=0.29
suic_plan_real	suic_plan_real=1.00, suic_plan_dtail=0.55, suic_prosp_leth=0.47, suic_steps_takn=0.42, suic_id_hi_risk=0.37, suic_pot_trig=0.34, suic_id_strngth=0.31, gen_paran_del_spec=0.29, gen_violent_purs=0.28, sh_pot_trigs_mtch=0.28
suic_plan_dtail	suic_plan_dtail=1.00, hto_to_anmls=0.60, suic_plan_real=0.55, suic_steps_takn=0.55, hto_means_plan=0.49, suic_prosp_leth=0.47, suic_id_hi_risk=0.47, suic_id_strngth=0.39, hto_fire_setting=0.34, suic_planning=0.34
suic_steps_takn	suic_steps_takn=1.00, family_ment_hlth=0.59, suic_plan_dtail=0.55, suic_id_hi_risk=0.46, suic_eol_prep=0.45, suic_plan_real=0.42, sh_planning=0.41, suic_id_control=0.40, suic_id_strngth=0.40, gen_decision=0.38
suic_prosp_leth	suic_prosp_leth=1.00, suic_plan_real=0.47, suic_plan_dtail=0.47, suic_id_hi_risk=0.42, suic_steps_takn=0.34, suic_lethality=0.32, suic_pot_trig=0.31, gen_paran_del_pers=0.31, suic_id_strngth=0.30, sh_pot_trigs_mtch=0.30
suic_eol_prep	suic_eol_prep=1.00, suic_steps_takn=0.45, suic_id_hi_risk=0.38, app_harm_dam=0.38, sh_planning=0.37, hto_steps_plan=0.37,

Node name	Related nodes
	suic_planning=0.35, suic_plan_dtail=0.33, hto_emotional_ep=0.32, suic_id_strngth=0.32
suic_pot_trig	suic_pot_trig=1.00, sh_pot_triggs=0.69, suic_p_trig_mtch=0.67, suic_id_hi_risk=0.60, suic_id_strngth=0.57, sh_pot_trigs_mtch=0.56, hto_pot_trig=0.53, suic_id_control=0.50, hto_pot_trig_mtch=0.48, sh_strength=0.47
suic_p_trig_mtch	suic_p_trig_mtch=1.00, suic_pot_trig=0.67, sh_pot_trigs_mtch=0.66, sh_pot_triggs=0.59, hto_pot_trig_mtch=0.58, suic_id_hi_risk=0.57, suic_id_strngth=0.56, suic_id_control=0.51, gen_life_not_livng=0.49, sh_strength=0.48
suic_id_control	suic_id_control=1.00, suic_id_strngth=0.77, suic_id_hi_risk=0.76, sh_strength=0.59, gen_prob_act_voice=0.57, gen_life_not_livng=0.55, suic_p_trig_mtch=0.51, suic_pot_trig=0.50, gen_voice_dang_s=0.49, gen_plans_future=0.48
suic_id_hi_risk	suic_id_hi_risk=1.00, suic_id_strngth=0.83, suic_id_control=0.76, gen_life_not_livng=0.62, gen_prob_act_voice=0.60, sh_strength=0.60, suic_pot_trig=0.60, suic_p_trig_mtch=0.57, gen_plans_future=0.54, gen_voice_dang_s=0.50
suic_id_strngth	suic_id_strngth=1.00, suic_id_hi_risk=0.83, suic_id_control=0.77, sh_strength=0.65, gen_life_not_livng=0.62, gen_voice_dang_s=0.58, suic_pot_trig=0.57, gen_prob_act_voice=0.56, suic_p_trig_mtch=0.56, gen_plans_future=0.52
gen_sh_cuts	gen_sh_cuts=1.00, sh=0.65, suic_s_h_behv=0.54, app_harm_dam=0.49, sh_strength=0.43, gen_prob_act_voice=0.42, hto_fam_hist=0.41, gen_voice_dang_s=0.40, gen_unint_risk_behavr=0.40, gen_negative_self=0.37
suic_s_h_behv	suic_s_h_behv=1.00, sh=0.62, gen_sh_cuts=0.54, hto_fam_hist=0.50, app_harm_dam=0.47, sh_lethality_mth=0.41, hto_steps_plan=0.40, gen_violent_purs=0.40, gen_unint_risk_behavr=0.40, gen_impulse=0.38
sh	sh=1.00, sh_strength=0.73, gen_sh_cuts=0.65, suic_s_h_behv=0.62, sh_pot_triggs=0.62, sh_pot_trigs_mtch=0.62, gen_voice_dang_s=0.53, hto_fam_hist=0.47, gen_prob_act_voice=0.46, suic_id_hi_risk=0.46
sh_planning	sh_planning=1.00, suic_planning=0.50, sh_lethality_mth=0.42, suic_steps_takn=0.41, sh=0.40, sh_hlp_after=0.40, suic_discovery=0.37, app_harm_dam=0.37, suic_eol_prep=0.37, sh_strength=0.36
sh_hlp_after	sh_hlp_after=1.00, sh_planning=0.40, suic_discovery=0.36, hto_to_anmls=0.35, suic_ser_succd=0.32, gen_eating_dis=0.30, suic_planning=0.28, gen_helpless=0.26, sh_strength=0.25, sh=0.25
sh_lethality_mth	sh_lethality_mth=1.00, sh_planning=0.42, suic_s_h_behv=0.41, suic_lethality=0.40, suic_planning=0.36, sh=0.33, suic_id_hi_risk=0.32, app_harm_dam=0.32, suic_id_strngth=0.32, sh_strength=0.31
sh_for_hlp_diff	sh_for_hlp_diff=1.00, gen_mood_swings=0.29, suic_id_control=0.29, suic_id_hi_risk=0.28, suic_s_h_behv=0.27, sh_lethality_mth=0.27, sh_pot_triggs=0.25, gen_sh_cuts=0.25, sh=0.25, sh_strength=0.25
sh_pot_triggs	sh_pot_triggs=1.00, sh_pot_trigs_mtch=0.77, suic_pot_trig=0.69, sh=0.62, hto_pot_trig=0.60, suic_p_trig_mtch=0.59, sh_strength=0.58, hto_pot_trig_mtch=0.55, gen_negative_self=0.47, suic_id_hi_risk=0.44

Table 80 Node to node relationship by snomed phrase

Node Name	Other node	Both present	Avg Risk	Chi	p-value	remark
gen_alc_misuse	gen_relat_detr_chg	13	3.23	13.74	0	increase
gen_alc_misuse	suic_lethality	16	4.31	9.12	0.003	increase
gen_alc_misuse	gen_life_not_livng	13	3.92	6.6	0.01	increase
gen_angry_emotns	hto_pot_trig	13	2.54	33.22	0	decrease
gen_angry_emotns	suic_pot_trig	35	4.43	42.97	0	increase
gen_angry_emotns	suic_id_control	15	4.87	16.92	0	increase
gen_angry_emotns	gen_mood_swings	40	3.58	115.24	0	increase
gen_angry_emotns	gen_negative_self	28	3.36	87.08	0	increase
gen_angry_emotns	gen_sad	17	4.18	37.97	0	increase
gen_angry_emotns	gen_life_not_livng	13	3.08	34.37	0	decrease
gen_angry_emotns	gen_sleep_dist	25	3.68	29.82	0	increase
gen_angry_emotns	gen_unusl_rec_bhvr	13	4.23	78.02	0	increase
gen_anx_emotns	sh_pot_triggs	21	3	25.7	0	decrease
gen_anx_emotns	gen_distress	18	2.56	61.46	0	decrease
gen_anx_emotns	gen_mood_swings	72	2.92	125.6	0	decrease
gen_anx_emotns	suic_ser_succd	15	4.6	5.05	0.025	increase
gen_anx_emotns	suic_p_trig_mtch	31	4.52	30.49	0	increase
gen_anx_emotns	suic_id_control	27	4.48	14.4	0	increase
gen_anx_emotns	suic_id_hi_risk	18	4.78	17.12	0	increase
gen_anx_emotns	gen_helpless	25	4.04	121.6	0	increase
gen_anx_emotns	gen_sad	42	3.4	97.37	0	increase
gen_anx_emotns	gen_life_not_livng	27	2.85	55.48	0	decrease
gen_anx_emotns	worthlessness	17	2.88	38.04	0	decrease
gen_anx_emotns	gen_phys_hlth_disa	20	3.45	15.33	0	increase
gen_anx_emotns	gen_phys_hlth_det	16	3.88	11.58	0.001	increase
gen_anx_emotns	gen_diet_eating	17	2.82	16.95	0	decrease
gen_anx_emotns	gen_env_grew_up	21	2.43	22.97	0	decrease
gen_app_diet	sn	62	3.19	36.75	0	increase
gen_app_diet	suic_pot_trig	50	4.26	9.03	0.003	increase
gen_app_diet	vuln_su	11	2.27	4.76	0.029	decrease
gen_app_diet	gen_chall_bhvr	11	3.09	9.53	0.002	increase
gen_app_diet	sh_pot_triggs	19	2.89	15.3	0	decrease
gen_app_diet	sn_recnt_app_chnge	11	2	30.01	0	decrease
gen_app_diet	gen_mood_swings	33	2.94	4.07	0.044	decrease
gen_app_diet	gen_negative_self	33	3.55	20.76	0	increase
gen_app_diet	gen_life_not_livng	20	3.9	19.15	0	increase
gen_app_diet	gen_phys_hlth_pain	23	3.61	6.46	0.011	increase

Node Name	Other node	Both present	Avg Risk	Chi	p-value	remark
gen_app_diet	gen_phys_hlth_disa	17	3.65	6.27	0.012	increase
gen_app_diet	gen_phys_hlth_det	19	2.37	17.1	0	decrease
gen_app_diet	gen_unint_risk_behavr	11	4.09	24.69	0	increase
gen_app_diet	gen_sleep_dist	44	3.73	17.99	0	increase
gen_app_diet	gen_diet_weigt_chg	13	2.46	17.66	0	decrease
gen_app_diet	gen_eating_dis	19	2.58	26.04	0	decrease
gen_chall_bhvr	hto_pot_trig	17	2.53	69.84	0	decrease
gen_chall_bhvr	hto	43	2.37	51.28	0	decrease
gen_chall_bhvr	gen_mood_swings	21	2.67	19.57	0	decrease
gen_chall_bhvr	gen_negative_self	11	3.64	5.45	0.02	increase
gen_chall_bhvr	gen_unint_risk_behavr	13	3.54	144.58	0	increase
gen_cog_think_mem	gen_phys_hlth_det	22	1.55	159.45	0	decrease
gen_cog_think_mem	hto	24	1.71	7.92	0.005	decrease
gen_cog_think_mem	gen_concentr	12	1.75	270.29	0	decrease
gen_cog_think_mem	gen_phys_hlth_pain	14	2.07	21.9	0	decrease
gen_com_imp	gen_phys_hlth_disa	26	2.31	377.85	0	decrease
gen_com_imp	gen_phys_hlth_det	12	2.42	88.45	0	decrease
gen_com_imp	gen_phys_hlth_pain	26	1.92	245.67	0	decrease
gen_concentr	hto	17	1.65	26.8	0	decrease
gen_coping_abil	suic_id_control	11	4.55	7	0.008	increase
gen_coping_abil	sh_pot_triggs	12	4.08	30.37	0	increase
gen_coping_abil	suic_p_trig_mtch	17	5.06	34.03	0	increase
gen_coping_abil	suic_lethality	12	4.67	23.73	0	increase
gen_coping_abil	risk_dep	14	4.21	22.92	0	increase

Table 81 Node to node relationship by node mg value

Node	Other node	Both count	Avg risk	Chi square	p-value	Remark
suic_discovery	suic_pot_trig	1157	5.57	33.32	0	increase
suic_discovery	suic_lethality	2050	4.47	562.43	0	increase
suic_lethality	sh_lethality_mth	1138	4.68	277.4	0	increase
suic_lethality	suic_pot_trig	2763	5.29	61.93	0	increase
suic_lethality	suic_p_trig_mtch	1804	5.61	34.96	0	increase
suic_lethality	sh_pot_triggs	1454	5.12	31.97	0	increase
suic_lethality	suic_s_h_behv	1078	5.13	90.12	0	increase
suic_ser_succd	sh_pot_triggs	1056	5.4	17.33	0	increase
suic_ser_succd	suic_pot_trig	2303	5.49	116.98	0	increase
suic_pot_trig	sh_pot_triggs	1878	5.15	698.04	0	increase

Node	Other node	Both count	Avg risk	Chi square	p-value	Remark
suic_pot_trig	hto_pot_trig	1013	5.06	308.58	0	increase
suic_pot_trig	sh_pot_trigs_mtch	1354	5.29	450.75	0	increase
suic_p_trig_mtch	sh_pot_triggs	1051	5.55	398.55	0	increase
hto_curr_persp_ep	gen_insght_behvr	1108	2.55	86.93	0	decrease
hto_curr_persp_ep	gen_resp_impct_oth	1332	2.85	159.92	0	decrease
carers	gen_liv_skills	1818	2.38	852.22	0	decrease
gen_mood_swings	suic_id_hi_risk	1055	6.35	219.6	0	increase
gen_negative_self	suic_id_hi_risk	1367	6.27	235.83	0	increase
gen_negative_self	suic_id_control	1089	6.27	214.35	0	increase
gen_helpless	suic_id_hi_risk	1278	6.35	448.03	0	increase
gen_helpless	suic_id_control	1005	6.33	322.86	0	increase
gen_plans_future	suic_id_hi_risk	1035	6.58	647.79	0	increase
gen_life_not_livng	suic_id_control	1026	6.49	621.96	0	increase
gen_life_not_livng	suic_id_hi_risk	1365	6.44	949.98	0	increase
gen_empathy_abil	gen_relat_supp	1379	3.68	100.41	0	increase
gen_coping_abil	suic_id_hi_risk	1007	6.3	39.86	0	increase
gen_coping_abil	suic_s_h_behv	1014	5.13	32.89	0	increase
gen_insght_behvr	gen_mentl_insght	1688	2.57	309.65	0	decrease
gen_nd_hlp_diff	gen_mentl_insght	1637	2.44	820.78	0	decrease
gen_net_relat	gen_relat_detr	2036	3.47	285.08	0	increase
gen_relat_supp	gen_relat_detr	1982	3.56	643.97	0	increase
gen_relat_supp	gen_neigbrhd_rsky	1056	3.55	156.97	0	increase
gen_alc_misuse	gen_drug_misuse	1166	3.54	232.52	0	increase
sh_for_hlp_diff	sh	1447	4.57	233.52	0	increase
sh_pot_trigs_mtch	sh	1846	4.64	849.03	0	increase
hto_pot_trig	hto	1620	3.56	699.12	0	increase
hto_pot_trig_mtch	hto	1100	3.43	555.38	0	increase
phys_vuln	vuln_su	1370	3.05	935.29	0	decrease
emot_vuln	vuln_su	1514	3.29	992.59	0	increase
carers	vuln_su	1301	2.59	243.61	0	decrease
suic_ser_succd	suic_regret	1939	5.19	576.46	0	increase
suic_ser_succd	suic_id_hi_risk	1049	6.6	265.5	0	increase
suic_ser_succd	suic_p_trig_mtch	1508	5.81	64.48	0	increase
suic_regret	suic_pot_trig	1639	5.71	206.43	0	increase
suic_regret	suic_p_trig_mtch	1142	5.91	214.96	0	increase
suic_pot_trig	vuln_su	1221	5.15	106.26	0	increase
suic_pot_trig	sh	1673	5.49	340.16	0	increase
suic_id_control	suic_pot_trig	1267	6.38	352.61	0	increase
suic_id_hi_risk	suic_pot_trig	1676	6.33	555.75	0	increase
suic_id_strngth	suic_pot_trig	1553	6.26	497.84	0	increase

Node	Other node	Both count	Avg risk	Chi square	p-value	Remark
sh_for_hlp_diff	sh_pot_triggs	1287	4.53	61.07	0	increase
sh_for_hlp_diff	suic_pot_trig	1121	5.24	22.95	0	increase
hto	vuln_su	1033	3.14	507.49	0	decrease
gen_mood_swings	suic_pot_trig	2391	5.27	259.59	0	increase
gen_mood_swings	sh_pot_triggs	1634	4.7	148.72	0	increase
gen_mood_swings	gen_alc_misuse	1085	4.31	9.03	0.003	increase
gen_negative_self	suic_pot_trig	2865	5.24	313.53	0	increase
gen_negative_self	sh_pot_triggs	1829	4.67	234.34	0	increase
gen_negative_self	vuln_su	1321	4.33	89.29	0	increase
gen_angry_emotns	suic_pot_trig	1235	5.28	96.47	0	increase

Table 82 Node node connections and correlation

nodecode	Othernode	Node distance	Common parentnode	Snomed match	Match any	Match other	Match aid	Corr.
suic_id_control	suic_id_strngth	2	suic_curr_sit_behav	16	14	14	10	0.78
suic_id_control	suic_id_hi_risk	2	suic_curr_sit_behav	16	8	9	9	0.74
gen_motivation	gen_listless	6	gen_direct	7	2	2	2	0.72
emot_vuln	phys_vuln	2	vuln_app_behavr	24	7	10	4	0.69
gen_helpless	gen_sad	2	gen_state_mind	22	5	5	5	0.66
gen_negative_self	Worthlessness	2	gen_state_mind	22	5	3	2	0.65
sh_pot_triggs	suic_pot_trig	8	mental_health_risk	62	8	6	3	0.62
suic_lethality	suic_ser_succd	4	suic_past_att	8	4	1	3	0.62
gen_distress	gen_helpless	2	gen_state_mind	19	2	2	2	0.6
suic_id_hi_risk	suic_pot_trig	2	suic_curr_sit_behav	20	4	6	3	0.59
gen_distress	gen_sad	2	gen_state_mind	24	6	3	2	0.58
suic_id_strngth	suic_pot_trig	2	suic_curr_sit_behav	19	1	1	5	0.56
hto_hi_rsk_ideatn	hto_pot_trig	2	harm_dam_ curr_sit_behav	6	2	2	2	0.56
hto	hto_pot_trig_mtch	6	root	25	4	4	2	0.56
emot_vuln	sex_vuln	2	vuln_app_behavr	19	4	5	2	0.55
gen_plans_future	gen_sad	3	gen_state_mind	17	1	2	2	0.54
sex_vuln	vuln_su	6	root	32	4	3	4	0.52
sn	vuln_su	2	root	105	17	14	4	0.52
gen_anx_emotns	gen_distress	2	gen_state_mind	21	5	5	5	0.52
gen_mania	gen_mood_swings	5	gen_direct	26	1	2	2	0.52
emot_vuln	finan_vuln	2	vuln_app_behavr	13	4	5	3	0.51
gen_life_not_livng	suic_id_control	7	suic	20	3	3	2	0.51
suic_id_control	suic_pot_trig	2	suic_curr_sit_behav	38	9	5	3	0.5

nodecode	Othernode	Node distance	Common parentnode	Snomed match	Match any	Match other	Match aid	Corr.
suic_id_hi_risk	suic_p_trig_mtch	2	suic_curr_sit_behav	18	4	4	2	0.5
gen_anx_emotns	gen_helpless	2	gen_state_mind	17	6	5	5	0.49
gen_plans_future	suic_id_hi_risk	7	suic	10	2	2	2	0.48
suic_id_hi_risk	suic_regret	5	sui_specific	9	1	7	2	0.48
gen_angry_emotns	Hto	6	root	36	5	5	4	0.47
finan_vuln	phys_vuln	2	vuln_app_behavr	15	4	6	2	0.45
suic_discovery	suic_lethality	2	suic_prep_serious_at	16	7	1	2	0.44
hto_pot_trig	suic_p_trig_mtch	8	mental_health_risk	28	3	3	2	0.44
hto_pot_trig	gen_prob_act_voice	11	mental_health_risk	7	1	1	4	0.43
gen_plans_future	suic_regret	8	suic	10	1	1	2	0.42
sn	gen_diet_eating	10	root	25	7	6	6	0.4
gen_anx_emotns	gen_sad	2	gen_state_mind	24	4	4	4	0.4
gen_coping_abil	gen_impulse	2	gen_person_thinking	10	5	4	2	0.4
finan_vuln	sex_vuln	2	vuln_app_behavr	11	2	4	2	0.39
gen_negative_self	suic_pot_trig	6	suic	40	4	3	2	0.38
gen_mood_swings	hto_pot_trig	8	mental_health_risk	34	9	5	5	0.37
gen_sad	suic_pot_trig	6	suic	43	3	2	5	0.37
gen_plans_future	suic_pot_trig	7	suic	37	1	1	2	0.37
gen_life_not_livng	sh_pot_triggs	9	mental_health_risk	22	1	1	2	0.37
suic_pot_trig	suic_regret	5	sui_specific	25	1	1	3	0.37
gen_sad	suic_id_hi_risk	6	suic	11	1	1	2	0.37
gen_distress	gen_mood_swings	2	gen_state_mind	22	4	4	2	0.37
hto	vuln_su	2	root	124	17	15	3	0.36
gen_distress	hto_pot_trig	8	mental_health_risk	22	1	1	3	0.36
gen_mood_swings	Sh	6	root	59	1	1	4	0.36
gen_mood_swings	suic_id_control	6	suic	27	6	5	2	0.35
gen_plans_future	suic_p_trig_mtch	7	suic	24	4	2	2	0.34
sh	suic_pot_trig	6	root	90	4	5	2	0.33
sn_recnt_app_chnge	gen_phys_hlth_det	8	suic	6	3	4	2	0.33
gen_helpless	gen_mood_swings	2	gen_state_mind	31	4	3	2	0.32
sn_recnt_app_chnge	gen_diet_eating	12	suic	8	2	2	2	0.31
suic_discovery	suic_regret	4	suic_past_att	12	3	1	4	0.31
sn	gen_phys_withd	9	root	20	2	2	2	0.31
gen_mood_swings	vuln_su	6	root	67	12	9	3	0.31
gen_empathy_abil	gen_mentl_insght	4	gen_direct	8	2	2	2	0.31
gen_distress	gen_prob_act_voice	7	gen_direct	7	3	3	3	0.3
gen_dependence	vuln_su	6	root	21	4	2	2	0.29
gen_distress	gen_voice_dang_s	8	gen_direct	7	3	3	3	0.29
gen_sad	suic_regret	7	suic	15	1	1	2	0.28
gen_voice_dang_s	hto_pot_trig	12	mental_health_risk	7	1	1	3	0.28
sh	vuln_su	2	root	119	12	14	3	0.27

nodecode	Othernode	Node distance	Common parentnode	Snomed match	Match any	Match other	Match aid	Corr.
gen_negative_self	suic_regret	7	suic	14	3	3	2	0.27
vuln_su	gen_phys_hlth_det	6	root	59	8	7	5	0.26
gen_mood_swings	suic_p_trig_mtch	6	suic	39	5	5	2	0.26
gen_coping_abil	Sh	6	root	23	5	4	3	0.26
suic_lethality	suic_pot_trig	5	sui_specific	22	3	1	5	0.25
suic_pot_trig	vuln_su	6	root	110	4	2	4	0.25
suic_pot_trig	gen_sleep_dist	9	suic	28	5	4	4	0.25
gen_helpless	vuln_su	6	root	47	2	2	2	0.24
gen_mood_swings	gen_sad	2	gen_state_mind	36	7	7	4	0.24
gen_angry_emotns	Sh	6	root	32	3	5	2	0.24
sh_lethality_mth	suic_p_trig_mtch	8	mental_health_risk	9	3	2	2	0.23
finan_vuln	gen_mood_swings	8	mental_health_risk	12	4	3	2	0.23
gen_anx_emotns	gen_mental_withd	7	gen_direct	13	4	4	4	0.23
hto_curr_persp_ep	vuln_su	5	root	30	4	3	3	0.22
hto	sh_pot_triggs	6	root	42	2	2	2	0.22
sn	gen_job_chg_frq	7	root	15	1	1	2	0.22
gen_mood_swings	risk_dep	6	root	37	2	2	2	0.22
gen_nd_hlp_diff	gen_meds_concord	8	suic	19	2	2	2	0.21
gen_cog_think_mem	vuln_su	7	root	35	4	2	2	0.21
gen_reliable	hto_pot_trig	8	mental_health_risk	12	5	4	4	0.2
sn	gen_alc_misuse	6	root	38	1	1	3	0.18
suic_pot_trig	gen_phys_hlth_pain	6	suic	36	2	3	4	0.17
gen_distress	hto_ideatn_link	8	mental_health_risk	6	1	1	2	0.17
gen_motivation	gen_mentl_insght	4	gen_health_care	7	2	2	2	0.17
gen_dependence	gen_relat_detr	6	gen_direct	9	4	4	4	0.16
sh_for_hlp_diff	suic_pot_trig	7	mental_health_risk	18	1	1	5	0.14
suic_regret	vuln_su	7	root	24	5	4	2	0.14
gen_angry_emotns	suic_regret	7	suic	8	3	3	2	0.14
gen_insght_behvr	gen_alc_misuse	8	suic	9	6	2	2	0.13
sh_for_hlp_diff	suic_lethality	8	mental_health_risk	10	5	5	5	0.13
hto	gen_alc_misuse	6	root	40	8	7	3	0.13
suic_pot_trig	gen_phys_hlth_disa	6	suic	27	1	1	2	0.12
sh	gen_drug_misuse	6	root	28	4	4	3	0.12
gen_mood_swings	gen_mentl_insght	4	gen_direct	28	3	3	2	0.12
suic_pot_trig	gen_mentl_insght	6	suic	33	1	1	2	0.11
gen_life_not_livng	Hto	7	root	20	6	1	2	0.11
gen_learn_disab	hto_curr_persp_ep	7	mental_health_risk	7	1	1	2	0.08
gen_coping_abil	gen_drug_misuse	4	gen_direct	8	9	3	3	0.07
gen_liv_skills	sh_pot_triggs	7	mental_health_risk	8	3	2	2	0.06
hto_curr_persp_ep	Sh	5	root	21	1	1	5	0.06
gen_insght_behvr	suic_lethality	7	sui_specific	9	2	2	2	0.01

nodecode	Othernode	Node distance	Common parentnode	Snomed match	Match any	Match other	Match aid	Corr.
gen_cog_think_mem	Sh	7	root	17	3	2	2	-0.03

Table 83 Grist node to snomed-ct concept mapping

Grist node name	Snomed-ct concept name
sn_appearnce	Victim of neglect
sn_hygiene	Smell
sn_recnt_app_chnge	Weight decreasing
sn_skin	Infective disorder
Suic	[X](Intentional self-harm) or (suicide) (event)
suic_curr_int	Thinking, function (observable entity)
suic_discovery	OD - Overdose of drug
suic_eol_prep	[X](Intentional self-harm) or (suicide) (event)
suic_escalate	[X](Intentional self-harm) or (suicide) (event)
suic_fam_hist	[X](Intentional self-harm) or (suicide) (event)
suic_fam_hist	[X](Intentional self-harm) or (suicide) (event)
suic_first_occ	[X](Intentional self-harm) or (suicide) (event)
suic_how_many	[X](Intentional self-harm) or (suicide) (event)
suic_id_control	Thinking, function (observable entity)
suic_id_freq	Thinking (observable entity)
suic_id_hi_risk	Thinking, function (observable entity)
suic_id_strngth	Thinking (observable entity)
suic_ideation	Thinking (observable entity)
suic_int_inform	[X](Intentional self-harm) or (suicide) (event)
suic_int_p_trig	[X](Intentional self-harm) or (suicide) (event)
suic_leth_insght	[X](Intentional self-harm) or (suicide) (event)
suic_lethality	(Poisoning (& [drug] &/or [biological substance] or [medicinal])) or (overdose: [biological substance] or [drug]) (disorder)

Grist node name	Snomed-ct concept name
suic_most_rec	OD - Overdose of drug
suic_note_prev	[X](Intentional self-harm) or (suicide) (event)
suic_p_trig_mtch	[X](Intentional self-harm) or (suicide) (event)
suic_past_att	[X](Intentional self-harm) or (suicide) (event)
suic_patt_att	(Poisoning (& [drug] &/or [biological substance] or [medicinal])) or (overdose: [biological substance] or [drug]) (disorder)
suic_plan_real	OD - Overdose of drug
suic_planning	[X](Intentional self-harm) or (suicide) (event)
suic_plans	Thinking, function (observable entity)
suic_pot_trig	Alcohol measurement
suic_regret	[X](Intentional self-harm) or (suicide) (event)
suic_rel_belief	Religious believer
suic_s_h_behv	[X](Intentional self-harm) or (suicide) (event)
suic_s_h_behv	[X](Intentional self-harm) or (suicide) (event)
suic_ser_succd	(Poisoning (& [drug] &/or [biological substance] or [medicinal])) or (overdose: [biological substance] or [drug]) (disorder)
suic_steps_takn	[X](Intentional self-harm) or (suicide) (event)
vuln_app_behavr	Abuse (event)
vuln_su	Abuse (event)
wandering	History of (contextual qualifier) (qualifier value)
worthlessness	Self-esteem

Appendix C **Risk Analysis Results**

Mallet: Following is the screenshot of the results given by Mallet tool for 10 classes prediction.

```
Trainer MaxEntTrainer, training data accuracy= 0.99
Trainer MaxEntTrainer, Test Data Confusion Matrix
Confusion Matrix, row=true, column=predicted  accuracy=0.29
  label    0    1    2    3    4    5    6    7    8    9   10  |total
  0  5,   84  125   65  113   91   34   38   16    3    2    .  |571
  1  3,  129  360  219  360  155   36   28   14    2    1    .  |1304
  2  1,   68  225  668  401   77    8   17   11    1    3    .  |1479
  3  2,  100  356  404  546  126   20   25    6    1    .    .  |1584
  4  4,   81  177   74  140  130   31   23    9    .    .    .  |665
  5  6,   45   66   26   46   47   25   22    9    2    .    .  |288
  6  7,   46   48   24   33   26   16   43   11    1    .    .  |248
  7  8,   26   21   11   16   20   19   18   24    5    .    .  |160
  8  9,    9    5    4    3    5    2   12    4    3    .    .  |47
  9 10,    .    .    .    2    1    4    2    5    1    .    .  |15
 10        .    .    .    .    .    .    .    .    .    .    .  |0


Trainer NaiveBayesTrainer training data accuracy= 0.589
Trainer NaiveBayesTrainer Test Data Confusion Matrix
Confusion Matrix, row=true, column=predicted  accuracy=0.316
  label    0    1    2    3    4    5    6    7    8    9   10  |total
  0  5,   50  228   65  167   47    9    5    .    .    .    .  |571
  1  3,   54  570  172  426   76    4    2    .    .    .    .  |1304
  2  1,   24  339  667  414   35    .    .    .    .    .    .  |1479
  3  2,   40  522  313  635   61    5    7    1    .    .    .  |1584
  4  4,   41  266   90  190   71    6    .    1    .    .    .  |665
  5  6,   42  117   21   69   27    6    6    .    .    .    .  |288
  6  7,   32   94   29   51   27    5   10    .    .    .    .  |248
  7  8,   23   63   18   28   20    4    2    2    .    .    .  |160
  8  9,    6   22    4    8    6    1    .    .    .    .    .  |47
  9 10,    1    6    .    5    1    1    1    .    .    .    .  |15
 10        .    .    .    .    .    .    .    .    .    .    .  |0
```

Figure 30 Mallet results for 10 category of risk

Mallet: Following is the screenshot of the results given by Mallet tool for 3 classes prediction.

```
Trainer MaxEntTrainer, training data accuracy= 0.99
Confusion Matrix, row=true, column=predicted  accuracy=0.69
 label    0    1    2  |total
  0 0, 3606 717   38  |4361
  1 1, 906 794   61  |1761
  2 2,  69 127   43  |239

Trainer NaiveBayesTrainer training data accuracy= 0.81
Confusion Matrix, row=true, column=predicted  accuracy=0.67
 label    0    1    2  |total
  0 0, 3309 1043   9  |4361
  1 1, 750 1009   2  |1761
  2 2,  54 178   7  |239
```

Figure 31 Mallet results for 3 category of risks

Stanford classifier: Following is the screenshot of Stanford classifier results.

```
- LinearClassifier with 121803 features, 10 classes, and 1218030 parameters.
- done [3.2s, 1001 items].
- Output format: dataColumn1   goldAnswer  classifierAnswer    P(clAnswer) P(goldAnswer)
-
- 1001 examples in test set
- Cls 7: TP=4 FN=25 FP=32 TN=940; Acc 0.943 P 0.111 R 0.138 F1 0.123
- Cls 1: TP=46 FN=118 FP=85 TN=752; Acc 0.797 P 0.351 R 0.280 F1 0.312
- Cls 3: TP=93 FN=173 FP=196 TN=539; Acc 0.631 P 0.322 R 0.350 F1 0.335
- Cls 2: TP=88 FN=157 FP=194 TN=562; Acc 0.649 P 0.312 R 0.359 F1 0.334
- Cls 4: TP=20 FN=94 FP=129 TN=758; Acc 0.777 P 0.134 R 0.175 F1 0.152
- Cls 6: TP=2 FN=50 FP=20 TN=929; Acc 0.930 P 0.091 R 0.038 F1 0.054
- Cls 5: TP=9 FN=77 FP=71 TN=844; Acc 0.852 P 0.113 R 0.105 F1 0.108
- Cls 8: TP=2 FN=27 FP=9 TN=963; Acc 0.964 P 0.182 R 0.069 F1 0.100
- Cls 10: TP=1 FN=4 FP=0 TN=996; Acc 0.996 P 1.000 R 0.200 F1 0.333
- Cls 9: TP=0 FN=11 FP=0 TN=990; Acc 0.989 P 1.000 R 0.000 F1 0.000
- Accuracy/micro-averaged F1: 0.26474
- Macro-averaged F1: 0.18520
```

Figure 32 Results from Stanford classifier

Experiment 1: Weka screenshot of snomed concepts extracted by cTAKES, 10 classes.

```
Correctly Classified Instances        1675        23.2348 %
Incorrectly Classified Instances      5534        76.7652 %
Kappa statistic                          0.0452
Mean absolute error                      0.1577
Root mean squared error                  0.3194
Relative absolute error                 96.3083 %
Root relative squared error            111.6278 %
Total Number of Instances             7209

Scheme:weka.classifiers.bayes.NaiveBayes
Relation:     snomed10c
Instances:    21203
Attributes:   1001
[list of attributes omitted]
Test mode:split 66.0% train, remainder test

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.689    0.546      0.27      0.689     0.388      0.612    1
                 0.171    0.15       0.277     0.171     0.211      0.52     2
                 0.088    0.07       0.246     0.088     0.13       0.531    3
                 0.052    0.038      0.147     0.052     0.077      0.546    4
                 0.051    0.031      0.136     0.051     0.074      0.538    5
                 0.039    0.025      0.066     0.039     0.049      0.542    6
                 0.044    0.031      0.057     0.044     0.05       0.554    7
                 0.035    0.039      0.022     0.035     0.027      0.54     8
                 0.053    0.015      0.028     0.053     0.036      0.609    9
                 0.067    0.01       0.014     0.067     0.024      0.464    10
Weighted Avg.    0.232    0.186      0.216     0.232     0.188      0.551

=== Confusion Matrix ===

    a    b    c    d    e    f    g    h    i    j   <-- classified as
 1125  210   89   40   36   33   35   36   13   15 |    a = 1
 1061  310  140   63   61   48   48   58   10   18 |    b = 2
  822  223  131   69   43   41   46   68   28   16 |    c = 3
  423  136   72   42   31   19   23   33   13    9 |    d = 4
  310   96   47   35   32   17   34   36   14    4 |    e = 5
  154   51   20   13   14   12   15   18    8    4 |    f = 6
  150   53   20   12   10    7   13   17    8    3 |    g = 7
   86   33    9   10    6    4    8    6   11    0 |    h = 8
   34    4    4    1    2    2    4    3    3    0 |    i = 9
    6    4    0    0    1    0    1    2    0    1 |    j = 10
```

Figure 33 Prediction by snomed 10 classes

Experiment 2: Weka screenshot snomed concepts extracted by cTAKES, 3 classes.

```
Correctly Classified Instances        4543        63.0184 %
Incorrectly Classified Instances      2666        36.9816 %
Kappa statistic                          0.0701
Mean absolute error                      0.2707
Root mean squared error                  0.4465
Relative absolute error                 90.0401 %
Root relative squared error            115.175  %
Total Number of Instances             7209

Scheme:weka.classifiers.bayes.NaiveBayes
Relation:     snomed3c
Instances:    21203
Attributes:   1001
[list of attributes omitted]
Test mode:split 66.0% train, remainder test

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.846     0.77       0.705     0.846     0.769      0.589    0
                 0.169     0.121      0.354     0.169     0.229      0.572    1
                 0.09      0.042      0.071     0.09      0.079      0.557    2
Weighted Avg.    0.63      0.563      0.585     0.63      0.594      0.583

=== Confusion Matrix ===

    a    b    c   <-- classified as
 4178  584  174 |    a = 0
 1569  343  116 |    b = 1
  181   42   22 |    c = 2
```

Figure 34 Prediction by snomed 3 classes

Experiment 3: Weka screenshot N-gram not filtered by ECM semantic 3 classes.

```
Correctly Classified Instances        4152        57.5947 %
Incorrectly Classified Instances      3057        42.4053 %
Kappa statistic                          0.0913
Mean absolute error                      0.2855
Root mean squared error                  0.5151
Relative absolute error                 94.9859 %
Root relative squared error            132.8733 %
Total Number of Instances             7209

Scheme:weka.classifiers.bayes.NaiveBayes
Relation:     phrase3nop
Instances:    21203
Attributes:   1000
[list of attributes omitted]
Test mode:split 66.0% train, remainder test

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.722     0.603      0.722     0.722     0.722      0.593    0
                 0.263     0.201      0.339     0.263     0.297      0.551    1
                 0.22      0.093      0.077     0.22      0.114      0.583    2
Weighted Avg.    0.576     0.472      0.593     0.576     0.582      0.581

=== Confusion Matrix ===

    a    b    c   <-- classified as
 3564  982  390 |    a = 0
 1237  534  257 |    b = 1
  133   58   54 |    c = 2
```

Figure 35 Prediction by ECM semantic

Experiment 4: Weka screenshot N-gram phrases filtered by ECM semantic 3 classes

```
Correctly Classified Instances        4279        59.3564 %
Incorrectly Classified Instances      2930        40.6436 %
Kappa statistic                          0.0963
Mean absolute error                      0.278
Root mean squared error                  0.493
Relative absolute error                 92.4638 %
Root relative squared error            127.1708 %
Total Number of Instances             7209

Scheme:weka.classifiers.bayes.NaiveBayes
Relation:     phrase3c
Instances:    21203
Attributes:   1001
[list of attributes omitted]
Test mode:split 66.0% train, remainder test

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.757     0.634      0.722     0.757     0.739      0.615    0
                 0.25      0.177      0.356     0.25      0.294      0.576    1
                 0.147     0.082      0.059     0.147     0.085      0.612    2
Weighted Avg.    0.594     0.487      0.596     0.594     0.591      0.604

=== Confusion Matrix ===

    a    b    c   <-- classified as
 3736  857  343 |    a = 0
 1293  507  228 |    b = 1
  149   60   36 |    c = 2
```

Figure 36 Prediction by ECM phraseness

Experiment 5: Weka screenshot snomed concepts compressed by String Stemming 3 classes.

```
Correctly Classified Instances        4421        61.3261 %
Incorrectly Classified Instances      2788        38.6739 %
Kappa statistic                          0.073
Mean absolute error                      0.2728
Root mean squared error                  0.4675
Relative absolute error                 90.7466 %
Root relative squared error            120.5757 %
Total Number of Instances             7209

Scheme:weka.classifiers.bayes.NaiveBayes
Relation:     phrase3con
Instances:    21203
Attributes:   1000
[list of attributes omitted]
Test mode:split 66.0% train, remainder test

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.806     0.725      0.707     0.806     0.753      0.585    0
                 0.202     0.152      0.342     0.202     0.254      0.558    1
                 0.143     0.051      0.09      0.143     0.111      0.585    2
Weighted Avg.    0.613     0.541      0.583     0.613     0.591      0.577

=== Confusion Matrix ===

    a    b    c   <-- classified as
 3977  750  209 |    a = 0
 1476  409  143 |    b = 1
  172   38   35 |    c = 2
```

Figure 37 Prediction by snomed string stemming

Experiment 6: Weka screenshot Snomed concepts compressed by Semantic Stemming 3 classes.

```
Correctly Classified Instances        4340               60.2025 %
Incorrectly Classified Instances      2869               39.7975 %
Kappa statistic                          0.1011
Mean absolute error                      0.2729
Root mean squared error                  0.4898
Relative absolute error                 90.7681 %
Root relative squared error            126.3485 %
Total Number of Instances             7209

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.772     0.645      0.722     0.772     0.746      0.618    0
                 0.239     0.17       0.355     0.239     0.285      0.576    1
                 0.18      0.075      0.078     0.18      0.108      0.636    2
Weighted Avg.    0.602     0.492      0.597     0.602     0.595      0.607

=== Confusion Matrix ===

    a    b    c   <-- classified as
 3812  823  301 |    a = 0
 1322  484  222 |    b = 1
  145   56   44 |    c = 2

Scheme:weka.classifiers.bayes.NaiveBayes
Relation:     phrase3vec
Instances:    21203
Attributes:   1001
[list of attributes omitted]
Test mode:split 66.0% train, remainder test
```

Figure 38 Prediction by snomed semantic stemming

```
Correctly Classified Instances        4771               66.1812 %
Incorrectly Classified Instances      2438               33.8188 %
Kappa statistic                          0.093
Mean absolute error                      0.2894
Root mean squared error                  0.3922
Relative absolute error                 96.2761 %
Root relative squared error            101.164  %
Total Number of Instances             7209

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.897     0.806      0.707     0.897     0.791      0.614    0
                 0.163     0.095      0.402     0.163     0.232      0.598    1
                 0.061     0.016      0.116     0.061     0.08       0.631    2
Weighted Avg.    0.662     0.579      0.601     0.662     0.609      0.61

=== Confusion Matrix ===

    a    b    c   <-- classified as
 4426  449   61 |    a = 0
 1645  330   53 |    b = 1
  188   42   15 |    c = 2
```

Figure 39 Sceenshot of prediction by document vector

```
Correctly Classified Instances        4836               67.0828 %
Incorrectly Classified Instances      2373               32.9172 %
Kappa statistic                          0.0866
Mean absolute error                      0.2518
Root mean squared error                  0.4134
Relative absolute error                 83.7675 %
Root relative squared error            106.6195 %
Total Number of Instances             7209

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.924     0.837       0.706     0.924      0.8        0.576     0
                 0.133     0.073       0.418     0.133      0.202      0.56      1
                 0.02      0.013       0.051     0.02       0.029      0.556     2
Weighted Avg.    0.671     0.594       0.602     0.671      0.606      0.571

=== Confusion Matrix ===

    a    b    c   <-- classified as
 4561  333   42 |    a = 0
 1706  270   52 |    b = 1
  197   43    5 |    c = 2
```

Figure 40 Sceenshot of Prediction by similarity to node

Table 84 Repeat assessment soft node data

Node Name	suic Increase	suic Decre.	suic same	Sub total	Has value	Corr.	change prob	Decr / Total
Dummy	925	4274	2141	4274	4274	1	1.00	1.00
Suic	925	1208	0	2133	4274	1	0.50	0.28
vuln_su	298	358	542	1198	3219	0.25	0.37	0.11
Sh	355	477	346	1178	3266	0.69	0.36	0.15
Hto	249	348	503	1100	2927	0.17	0.38	0.12
Sn	214	292	446	952	2597	0.26	0.37	0.11
gen_distress	202	283	294	779	1957	0.43	0.40	0.14
gen_sad	214	282	279	775	2027	0.52	0.38	0.14
gen_anx_emotns	181	258	259	698	1947	0.25	0.36	0.13
gen_mentl_insght	165	195	273	633	2650	-0.01	0.24	0.07
gen_helpless	176	240	217	633	1866	0.54	0.34	0.13
suic_pot_trig	210	242	180	632	1730	0.46	0.37	0.14
gen_mood_swings	185	215	209	609	2007	0.42	0.30	0.11
gen_life_not_livng	169	253	184	606	1760	0.63	0.34	0.14

Node Name	suic Increase	suic Decre.	suic same	Sub total	Has value	Corr.	change prob	Decr / Total
gen_negative_self	161	227	188	576	1915	0.49	0.30	0.12
gen_plans_future	160	223	183	566	1786	0.49	0.32	0.12
gen_angry_emotns	139	200	220	559	1690	0.25	0.33	0.12
gen_impulse	138	167	136	441	2561	0.19	0.17	0.07
gen_alc_misuse	144	128	157	429	2000	0.29	0.21	0.06
suic_p_trig_mtch	143	162	106	411	1279	0.6	0.32	0.13
gen_hostile	107	124	143	374	2136	0.11	0.18	0.06
gen_mania	75	103	193	371	1300	0.11	0.29	0.08
gen_sleep_dist	80	115	148	343	1309	0.28	0.26	0.09
gen_dependence	89	130	107	326	2503	0.09	0.13	0.05
gen_insght_behvr	90	86	147	323	2181	0.11	0.15	0.04
gen_day_struct	82	87	148	317	1311	0.16	0.24	0.07
suic_regret	107	114	93	314	2329	0.33	0.13	0.05
gen_empathy_abil	90	99	118	307	2487	0.07	0.12	0.04
emot_vuln	67	76	163	306	1204	0.22	0.25	0.06
gen_reliable	97	94	114	305	2364	0.05	0.13	0.04
sh_pot_triggs	83	121	95	299	1057	0.46	0.28	0.11
gen_nd_hlp_diff	66	79	152	297	2208	-0.07	0.13	0.04
gen_coping_abil	102	107	84	293	2590	0.29	0.11	0.04
gen_net_relat	94	97	100	291	2716	0.27	0.11	0.04
hto_pot_trig	67	84	140	291	1003	0.33	0.29	0.08
gen_resp_impct_oth	80	82	125	287	2296	0.3	0.13	0.04
gen_chall_bhvr	59	90	134	283	1194	0.17	0.24	0.08
gen_relat_supp	76	102	104	282	2679	0.2	0.11	0.04
worthlessness	93	103	84	280	1014	0.47	0.28	0.10
phys_vuln	55	79	145	279	1128	0.11	0.25	0.07
sn_hair_clothes	60	77	141	278	935	0.12	0.30	0.08
gen_drug_misuse	88	92	93	273	1389	0.37	0.20	0.07
sn_hygiene	49	84	135	268	904	0.12	0.30	0.09
gen_unusl_rec_bhvr	64	88	113	265	1091	0.23	0.24	0.08
suic_id_strngth	86	106	69	261	949	0.78	0.28	0.11
suic_id_control	85	99	73	257	945	0.65	0.27	0.10
suic_id_hi_risk	77	94	64	235	935	0.86	0.25	0.10
hto_pot_trig_mtch	55	66	110	231	796	0.31	0.29	0.08
gen_relat_detr_chg	58	89	79	226	1080	0.24	0.21	0.08
sh_pot_trigs_mtch	59	91	75	225	867	0.52	0.26	0.10
gen_listless	48	75	90	213	836	0.22	0.25	0.09
gen_diet_eating	51	73	87	211	878	0.22	0.24	0.08
gen_mental_withd	51	76	81	208	862	0.23	0.24	0.09

Node Name	suic Increase	suic Decre.	suic same	Sub total	Has value	Corr.	change prob	Decr / Total
suic_ser_succd	83	69	52	204	2374	0.53	0.09	0.03
gen_unint_risk_behavr	46	62	91	199	1061	0.21	0.19	0.06
gen_sh_cuts	53	68	77	198	803	0.47	0.25	0.08
sh_lethality_mth	74	67	57	198	1902	0.33	0.10	0.04
gen_motivation	48	74	74	196	849	0.31	0.23	0.09
gen_meds_concord	41	53	94	188	755	0.26	0.25	0.07
finan_vuln	45	50	91	186	953	0.2	0.20	0.05
sex_vuln	38	49	96	183	919	0.14	0.20	0.05
suic_lethality	87	49	46	182	2593	0.38	0.07	0.02
hto_curr_persp_ep	53	45	83	181	1602	0.2	0.11	0.03
risk_dep	44	73	63	180	872	0.39	0.21	0.08
suic_leth_insght	64	56	59	179	2016	0.34	0.09	0.03
sn_recnt_app_chnge	35	54	80	169	712	0.2	0.24	0.08
gen_rsk_behavr	41	46	80	167	941	0.3	0.18	0.05
sn_skin	36	57	71	164	739	0.15	0.22	0.08
gen_phys_hlth_det	19	55	90	164	656	0.24	0.25	0.08
sh_for_hlp_diff	60	55	38	153	1849	0.32	0.08	0.03
gen_relat_detr	57	52	43	152	1986	0.25	0.08	0.03
gen_phys_withd	40	52	60	152	787	0.2	0.19	0.07
suic_discovery	62	57	27	146	2223	0.5	0.07	0.03
carers	43	30	67	140	1409	-0.05	0.10	0.02
gen_med_perc_benft	39	41	57	137	622	0.24	0.22	0.07
gen_liv_skills	32	27	77	136	1301	-0.18	0.10	0.02
app_vuln_abuse	25	41	68	134	735	0.08	0.18	0.06
gen_paran_del_pers	31	29	69	129	544	0.15	0.24	0.05
gen_prob_act_par_del	20	33	75	128	515	0.38	0.25	0.06
sh_hlp_after	47	34	44	125	1509	0.25	0.08	0.02
gen_accom_habitbl	22	43	57	122	680	0.12	0.18	0.06
gen_serv_perc_supp	31	39	51	121	688	0.18	0.18	0.06
gen_phys_hlth_pain	32	37	51	120	1221	0.32	0.10	0.03
gen_neigbrhd_rsky	38	43	35	116	1582	0.07	0.07	0.03
gen_concentr	19	36	59	114	569	0	0.20	0.06
gen_paran_del_spec	28	29	55	112	497	0.28	0.23	0.06
gen_phys_hlth_disa	22	36	51	109	1234	0.11	0.09	0.03
gen_voice_dang_s	27	39	42	108	448	0.63	0.24	0.09
gen_diet_weigt_chg	27	30	50	107	545	0.1	0.20	0.06
gen_jealous	25	38	43	106	804	0.14	0.13	0.05
gen_prob_act_voice	28	28	48	104	461	0.55	0.23	0.06
gen_diet_drink	17	34	50	101	526	0.15	0.19	0.06

Node Name	suic Increase	suic Decre.	suic same	Sub total	Has value	Corr.	change prob	Decr / Total
gen_accom_hm_care	20	35	45	100	558	0.12	0.18	0.06
gen_perc_debt_anx	32	31	37	100	619	0.24	0.16	0.05
suic_planning	34	32	19	85	873	0.55	0.10	0.04
gen_cog_think_mem	18	25	38	81	493	-0.04	0.16	0.05
suic_s_h_behv	40	23	15	78	1606	0.58	0.05	0.01
hto_violent	22	20	32	74	1474	-0.03	0.05	0.01
sh_strength	20	31	22	73	385	0.57	0.19	0.08
gen_congruence	18	21	32	71	520	0.08	0.14	0.04
grandiosity	18	22	17	57	397	0.09	0.14	0.06
dis_conf	11	15	30	56	351	0.14	0.16	0.04
gen_poverty	15	21	19	55	892	0.16	0.06	0.02
gen_env_grew_up	20	20	15	55	2020	0.31	0.03	0.01
sh_planning	26	13	16	55	638	0.27	0.09	0.02
suic_eol_prep	21	19	12	52	378	0.36	0.14	0.05
gen_voice_dang_o	16	18	18	52	318	0.23	0.16	0.06
gen_isol_accom	12	14	21	47	1279	0.12	0.04	0.01
hto_dest_prprty	7	17	19	43	925	0.05	0.05	0.02
gen_rec_bad_job_ch	12	19	9	40	305	0.2	0.13	0.06
gen_risk_upbeat	12	11	16	39	322	0.51	0.12	0.03
gen_risk_aggrsv	10	9	19	38	287	-0.22	0.13	0.03
gen_com_imp	8	16	12	36	557	0.01	0.06	0.03
gen_detached	6	8	20	34	229	0.12	0.15	0.03
gen_responsve	6	10	17	33	246	-0.01	0.13	0.04
gen_distrss_b_lang	7	11	14	32	243	0.28	0.13	0.05
gen_coherence	8	5	18	31	265	-0.14	0.12	0.02
gen_educ_expr	9	9	13	31	1506	0.27	0.02	0.01
gen_avoid_eye_contact	4	9	16	29	214	0.16	0.14	0.04
gen_eating_dis	9	14	5	28	942	0.2	0.03	0.01
suic_prosp_leth	7	12	8	27	240	0.3	0.11	0.05
gen_low_mood	5	11	11	27	230	0.45	0.12	0.05
gen_rapport	5	10	12	27	246	0.04	0.11	0.04
gen_job_chg_frq	10	7	10	27	774	0.12	0.03	0.01
suic_plan_real	5	11	10	26	281	0.41	0.09	0.04
suic_steps_takn	7	8	7	22	230	0.38	0.10	0.03
hto_emotional_ep	6	8	7	21	486	0.17	0.04	0.02
gen_gut_assmnt	4	7	10	21	222	0.1	0.09	0.03
hto_hi_rsk_ideatn	5	7	8	20	157	0.35	0.13	0.04
gen_violent_purs	4	7	9	20	426	0.29	0.05	0.02
hto_ideatn_link	6	4	10	20	145	0.28	0.14	0.03

Node Name	suic Increase	suic Decre.	suic same	Sub total	Has value	Corr.	change prob	Decr / Total
hto_means_plan	6	5	8	19	151	0.08	0.13	0.03
hto_steps_plan	6	4	9	19	125	0.06	0.15	0.03
gen_threat_move	2	4	13	19	179	-0.18	0.11	0.02
hto_strgth_ideatn	5	8	5	18	119	0.41	0.15	0.07
suic_plan_dtail	6	7	2	15	109	0.42	0.14	0.06
gen_decision	1	3	10	13	34	-0.16	0.38	0.09
hto_fam_hist	3	3	6	12	480	0.44	0.03	0.01
app_harm_dam	3	3	4	10	98	0.39	0.10	0.03
gen_learn_disab	2	2	5	9	234	0.05	0.04	0.01
hto_fire_setting	2	3	4	9	321	0.17	0.03	0.01
hto_to_anmls	1	1	1	2	50	-0.17	0.04	0.02

Table 85 GRiST node and Information Gain

Node name	Corr. with suicide	Information Gain	GainRatio
suic_answer	1	2.6	0
gen_app_diet_answer	0	1.92	0.828
suic_id_hi_risk_answer	0.86	0.87	0.292
suic_id_strngth_answer	0.78	0.71	0.241
suic_id_control_answer	0.65	0.51	0.181
suic_pot_trig_answer	0.46	0.39	0.132
suic_p_trig_mtch_answer	0.6	0.38	0.12
sh_answer	0.69	0.32	0.119
app_harm_dam_answer	0.39	0.25	0.117
gen_life_not_livng_answer	0.63	0.33	0.111
sh_strength_answer	0.57	0.3	0.1
hto_to_anmls_answer	-0.17	0.26	0.086
gen_voice_dang_s_answer	0.63	0.25	0.082
gen_prob_act_voice_answer	0.55	0.24	0.08
suic_plan_real_answer	0.41	0.23	0.073
suic_steps_takn_answer	0.38	0.22	0.068
sh_pot_triggs_answer	0.46	0.2	0.067
suic_s_h_behv_answer	0.58	0.19	0.065
gen_low_mood_answer	0.45	0.2	0.064
sh_pot_trigs_mtch_answer	0.52	0.19	0.063

Node name	Corr. with suicide	Information Gain	GainRatio
gen_plans_future_answer	0.49	0.18	0.062
gen_risk_upbeat_answer	0.51	0.18	0.056
suic_regret_answer	0.33	0.17	0.055
family_ment_hlth_answer	0.08	0.18	0.055
hto_steps_plan_answer	0.06	0.18	0.054
suic_prosp_leth_answer	0.3	0.16	0.053
hto_means_plan_answer	0.08	0.17	0.053
gen_sh_cuts_answer	0.47	0.15	0.052
worthlessness_answer	0.47	0.15	0.048
hto_strgth_ideatn_answer	0.41	0.15	0.048
hto_fam_hist_answer	0.44	0.14	0.047
suic_eol_prep_answer	0.36	0.13	0.047
hto_hi_rsk_ideatn_answer	0.35	0.14	0.046
gen_helpless_answer	0.54	0.14	0.044
gen_sad_answer	0.52	0.14	0.044
gen_negative_self_answer	0.49	0.14	0.044
hto_fire_setting_answer	0.17	0.13	0.043
gen_voice_dang_o_answer	0.23	0.1	0.04
hto_emotional_ep_answer	0.17	0.13	0.04
hto_pot_trig_answer	0.33	0.12	0.039
gen_violent_purs_answer	0.29	0.09	0.038
sh_planning_answer	0.27	0.09	0.038
suic_ser_succd_answer	0.53	0.12	0.037
hto_pot_trig_mtch_answer	0.31	0.11	0.037
gen_gut_assmnt_answer	0.1	0.11	0.037
gen_distrss_b_lang_answer	0.28	0.12	0.035
hto_ideatn_link_answer	0.28	0.11	0.034
suic_planning_answer	0.55	0.09	0.031
gen_distress_answer	0.43	0.1	0.031
gen_impulse_answer	0.19	0.1	0.031
gen_threat_move_answer	-0.18	0.09	0.031
risk_dep_answer	0.39	0.07	0.03
gen_coherence_answer	-0.14	0.09	0.03
gen_prob_act_par_del_answer	0.38	0.09	0.029
gen_risk_aggrsv_answer	-0.22	0.09	0.028
hto_answer	0.17	0.07	0.027
gen_mood_swings_answer	0.42	0.08	0.026
suic_lethality_answer	0.38	0.08	0.026
sh_lethality_mth_answer	0.33	0.08	0.026
gen_motivation_answer	0.31	0.07	0.023

Node name	Corr. with suicide	Information Gain	GainRatio
gen_listless_answer	0.22	0.07	0.023
suic_discovery_answer	0.5	0.06	0.022
sn_answer	0.26	0.06	0.022
grandiosity_answer	0.09	0.05	0.022
suic_leth_insght_answer	0.34	0.06	0.02
gen_paran_del_pers_answer	0.15	0.06	0.02
dis_conf_answer	0.14	0.06	0.02
gen_detached_answer	0.12	0.07	0.02
gen_learn_disab_answer	0.05	0.07	0.02
gen_decision_answer	-0.16	0.07	0.02
gen_coping_abil_answer	0.29	0.05	0.019
gen_paran_del_spec_answer	0.28	0.06	0.019
gen_rapport_answer	0.04	0.06	0.019
gen_relat_detr_chg_answer	0.24	0.06	0.018
gen_perc_debt_anx_answer	0.24	0.05	0.018
gen_mental_withd_answer	0.23	0.06	0.018
gen_unint_risk_behavr_answer	0.21	0.05	0.018
sn_recnt_app_chnge_answer	0.2	0.06	0.018
gen_avoid_eye_contact_answer	0.16	0.06	0.018
sex_vuln_answer	0.14	0.04	0.017
app_vuln_abuse_answer	0.08	0.04	0.017
emot_vuln_answer	0.22	0.05	0.016
sh_for_hlp_diff_answer	0.32	0.05	0.015
gen_rec_bad_job_ch_answer	0.2	0.05	0.015
gen_jealous_answer	0.14	0.03	0.015
gen_rsk_behavr_answer	0.3	0.04	0.014
gen_sleep_dist_answer	0.28	0.05	0.014
sh_hlp_after_answer	0.25	0.04	0.014
gen_unusl_rec_bhvr_answer	0.23	0.05	0.014
sn_skin_answer	0.15	0.04	0.014
gen_diet_weigt_chg_answer	0.1	0.05	0.014
gen_reliable_answer	0.05	0.04	0.014
gen_nd_hlp_diff_answer	-0.07	0.04	0.014
gen_phys_hlth_pain_answer	0.32	0.04	0.013
gen_meds_concord_answer	0.26	0.04	0.013
vuln_su_answer	0.25	0.04	0.013
gen_med_perc_benft_answer	0.24	0.04	0.013
gen_diet_eating_answer	0.22	0.04	0.013
gen_eating_dis_answer	0.2	0.04	0.013
gen_poverty_answer	0.16	0.04	0.013

Node name	Corr. with suicide	Information Gain	GainRatio
gen_congruence_answer	0.08	0.04	0.013
gen_responsve_answer	-0.01	0.04	0.013
gen_liv_skills_answer	-0.18	0.04	0.013
finan_vuln_answer	0.2	0.03	0.012
gen_diet_drink_answer	0.15	0.04	0.012
gen_com_imp_answer	0.01	0.02	0.012
gen_phys_hlth_det_answer	0.24	0.04	0.011
phys_vuln_answer	0.11	0.03	0.011
gen_concentr_answer	0	0.03	0.011
carers_answer	-0.05	0.03	0.011
gen_phys_withd_answer	0.2	0.03	0.01
sn_hair_clothes_answer	0.12	0.03	0.01
gen_accom_habitbl_answer	0.12	0.03	0.01
gen_mania_answer	0.11	0.03	0.01
gen_cog_think_mem_answer	-0.04	0.03	0.01
gen_resp_impct_oth_answer	0.3	0.03	0.009
gen_angry_emotns_answer	0.25	0.02	0.009
gen_anx_emotns_answer	0.25	0.03	0.009
hto_curr_persp_ep_answer	0.2	0.03	0.009
gen_serv_perc_supp_answer	0.18	0.03	0.009
sn_hygiene_answer	0.12	0.03	0.009
gen_accom_hm_care_answer	0.12	0.02	0.009
gen_job_chg_frq_answer	0.12	0.02	0.009
gen_neigbrhd_rsky_answer	0.07	0.03	0.009
hto_dest_prprty_answer	0.05	0.03	0.009
gen_mentl_insght_answer	-0.01	0.03	0.009
gen_drug_misuse_answer	0.37	0.02	0.008
gen_env_grew_up_answer	0.31	0.03	0.008
gen_relat_supp_answer	0.2	0.03	0.008
gen_day_struct_answer	0.16	0.02	0.008
gen_isol_accom_answer	0.12	0.02	0.008
hto_violent_answer	-0.03	0.02	0.008
gen_alc_misuse_answer	0.29	0.02	0.007
gen_net_relat_answer	0.27	0.03	0.007
gen_chall_bhvr_answer	0.17	0.02	0.007
gen_insght_behvr_answer	0.11	0.02	0.007
gen_dependence_answer	0.09	0.02	0.007
gen_empathy_abil_answer	0.07	0.02	0.007
gen_relat_detr_answer	0.25	0.01	0.006