UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Unsupervised Discovery of Relations for Analysis of Textual Data in Digital Forensics

by

Anita Lily Louis

Submitted in partial fulfillment of the requirements for the degree
Master of Science (Computer Science)
in the Faculty of Engineering, Built Environment and Information Technology
University of Pretoria, Pretoria

August 2009

Publication data:

Anita Lily Louis. Unsupervised Discovery of Relations for Analysis of Textual Data in Digital Forensics. Master's dissertation, University of Pretoria, Department of Computer Science, Pretoria, South Africa, August 2009.

Electronic, hyperlinked versions of this dissertation are available online, as Adobe PDF files, at:

http://cirg.cs.up.ac.za/

http://upetd.up.ac.za/UPeTD.htm

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Unsupervised Discovery of Relations for Analysis of Textual Data in Digital Forensics

by

Anita Lily Louis
E-mail: alouis@csir.co.za

Abstract

This dissertation addresses the problem of analysing digital data in digital forensics. It will be shown that text mining methods can be adapted and applied to digital forensics to aid analysts to more quickly, efficiently and accurately analyse data to reveal truly useful information.

Investigators who wish to utilise digital evidence must examine and organise the data to piece together events and facts of a crime. The difficulty with finding relevant information quickly using the current tools and methods is that these tools rely very heavily on background knowledge for query terms and do not fully utilise the content of the data.

A novel framework in which to perform evidence discovery is proposed in order to reduce the quantity of data to be analysed, aid the analysts' exploration of the data and enhance the intelligibility of the presentation of the data. The framework combines information extraction techniques with visual exploration techniques to provide a novel approach to performing evidence discovery, in the form of an evidence discovery system. By utilising unrestricted, unsupervised information extraction techniques, the investigator does not require input queries or keywords for searching, thus enabling the investigator to analyse portions of the data that may not have been identified by keyword searches.

The evidence discovery system produces text graphs of the most important concepts and associations extracted from the full text to establish ties between the concepts and provide an overview and general representation of the text. Through an interactive

visual interface the investigator can explore the data to identify suspects, events and the relations between suspects.

Two models are proposed for performing the relation extraction process of the evidence discovery framework. The first model takes a statistical approach to discovering relations based on co-occurrences of complex concepts. The second model utilises a linguistic approach using named entity extraction and information extraction patterns.

A preliminary study was performed to assess the usefulness of a text mining approach to digital forensics as against the traditional information retrieval approach. It was concluded that the novel approach to text analysis for evidence discovery presented in this dissertation is a viable and promising approach. The preliminary experiment showed that the results obtained from the evidence discovery system, using either of the relation extraction models, are sensible and useful. The approach advocated in this dissertation can therefore be successfully applied to the analysis of textual data for digital forensics.

Keywords: Digital forensics, text analysis, text mining, information extraction, relation discovery.

Supervisor : Prof. A. Engelbrecht
Department : Department of Computer Science
Degree : Master of Science

"A criminal is a person with predatory instincts who has not sufficient capital to form a corporation."

Howard Scott (1926 - present)

"I took a speed reading course and read 'War and Peace' in twenty minutes. It involves Russia."

Woody Allen (1935 - present)

Acknowledgements

I would like to take this opportunity to thank the following people and organisations who were instrumental in the completion of the dissertation:

- My loving husband, friend and editor, Kevin, for his continuing support, encouragement and understanding. In addition for proof reading and correcting my grammar without complaint, and learning more about the field of computer science than he probably ever wanted to know.
- My supervisor Prof. Andries Engelbrecht for his guidance and academic support.
- Dr. Thomas Meyer and the members of the Knowledge Systems Group (KSG) at the CSIR, Meraka Institute for their encouragement, support and belief in my work.
- The Council for Scientific and Industrial Research (CSIR), The National Research Foundation (NRF), and the University of Pretoria (UP) for their financial contributions, without which this research would not have been possible. The opinions expressed in this work and the conclusions arrived at are my own and should not necessarily be attributed to the CSIR, NRF, or the UP.
- Prof. Grishman and Prof. Sekine at the Natural Language Processing research group at New York University for sharing their knowledge and advice, and for hosting me at their university.
- The volunteers who participated in the experiment in this work, for giving their valuable time.

Contents

List of Figures

List of Graphs

List of Algorithms

List of Tables

Chapter 1

Introduction

The use of the many forms of computer and communication devices today results in the generation of enormous pools of data which, in turn, enables criminal investigators to make use of this data to uncover evidence. A great deal of digital data is linguistic in nature (e.g. human languages, programming languages, and system and application logging conventions) (Beebe, 2007). The expressivity of language makes these files rich information sources, which makes textual data and consequently textual evidence very important in digital investigations.

The obvious and immediate difficulty which confronts any person wishing to make use of the potential which digital evidence holds, is that analysing and processing the ever-growing volumes of data, and linking the unstructured data together for use as evidence in a trial involves a tremendous amount of work. Examining and organising all of the data which digital evidence yields is a major challenge facing all investigators. For electronic evidence to become properly useful to investigators, accurate, efficient and rapid means of data analysis needs to be developed (Chen *et al.*, 2004).

1.1 Motivation

Computer forensic software exists that can assist in many phases of a digital investigation, however it is important to note that this software is not designed to solve crime. The primary goal of these systems is to reduce investigation time and complexity (Abraham,

2006).

Expression based search methods currently dominate the software tools used for the analysis of forensic data (Beebe, 2007; AccessData, 2008; Carrier, 2008; GuidanceSoftware, 2002). These tools focus on providing advanced search technologies with which an investigator can retrieve potential evidence. Thus search technology is called information retrieval. An investigator will use his/her experience and background knowledge of the case to select appropriate search terms and criteria to find clues in the data as to which data should be investigated in more detail, or to find data which will offer suggestions as to how to proceed with the investigation. However, it is not uncommon in digital investigations for very little to be known about the case or the collected data prior to analysis.

Information retrieval (IR) is only as good as the query terms used, which means that the evidence retrieved in a digital investigation is limited to the background knowledge of the case and extended search terms from the investigator's personal experience. Arguably the dependence on query terms and their many combinations makes searching for digital evidence very time consuming and inefficient.

A move away from information retrieval techniques includes text summarisation, document classification or clustering and text mining (Hotho *et al.*, 2005; Dozier and Jackson, 2005; Fan *et al.*, 2006; de Waal *et al.*, 2008). In a perfect world, a forensic system could "discover" data that seems suspicious. A discovery system which does not rely on user input for query terms could therefore assist in speeding up the analysis phase of the investigation. This has led researchers to investigate different approaches to finding evidence.

Promising results have been achieved by text mining in the biosciences application area. Perhaps the most cited example is Don Swanson's work on hypothesizing causes of rare diseases by looking for indirect links in different subsets of the bioscience literature (Swanson, 1987, 1991).

Dr. Liebman is convinced that new cures could someday emerge for breast cancer if only someone could read all the literature and synthesize it (Guernsey, 2003). Thus, Dr. Liebman enlisted the use of text mining software to 'read' medical journal articles. The preparation for Dr. Liebman's project took months to build a framework of knowledge

on the subject of interest (breast cancer), required as input to the system (Guernsey, 2003). The output of the software is a visual map of extracted concepts, which lead Dr. Liebman and his team down new pathways which they could then test scientifically.

The promising results of text mining in the biosciences field provides the motivation to explore the application of text mining methods as an alternative approach to finding evidence in digital forensics.

The problem of analysing digital data in digital forensics is addressed in this dissertation. It will be shown that text mining methods can be adapted and applied to digital forensics to aid analysts to more quickly, efficiently and accurately analyse data to reveal truly useful information.

1.2 Scope

The focus of this work is on the analysis of textual digital data for digital forensics. A large portion of the data that need to be analysed is in textual format and of a linguistic nature. Data that is in a regularised format such as a database, or financial data in a spread sheet can be more easily processed than data represented in natural language. The challenges that linguistic data presents provide large scope for research and investigation. Thus the scope of this research is limited to the analysis of textual linguistic data.

1.3 Objectives

The main objective of this research is to investigate the adaptation and application of text mining methods to the analysis of textual data for the purposes of digital forensic investigation. To address this main objective, the following sub-objectives have been identified:

- Identify the requirements and needs for advancement in the analysis phase of a digital forensic investigation.
- Conduct a survey of available techniques in the field of text mining.

- Investigate the adaptation and application of text mining research to digital forensics to address the needs identified for the analysis of textual data in digital forensics.
- Assess, in at least preliminary form, the usefulness of a text mining research approach to digital forensics as an alternative to the traditional IR approach.

1.4 Contributions

It is hypothesized in this dissertation that information extraction techniques combined with visual exploration techniques can assist in identifying suspects and events, and the relations between these entities which could assist an investigator to: piece together the story surrounding a crime, create hypotheses or potential leads for further investigation, or identify pieces of data which could form useful supporting evidence in a trial. This research investigates the adaptation and application of text mining research to digital forensics to address the need for a tool to aid analysts to more quickly, efficiently and accurately analyse data to reveal truly useful information.

The primary contributions of this work are:

- a novel framework in which to perform evidence discovery,
- a statistically based model for performing relation extraction (model A), and
- an alternative linguistically based model for performing relation extraction (model B) as components for the evidence discovery framework.

A preliminary study was devised to to assess the usefulness of a text mining approach to digital forensics as against the traditional information retrieval approach, untilising Agatha Christie's novel entitled 'The Mysterious Affair at Styles' as a dataset. The preliminary experiment showed that the approach advocated in this thesis can therefore be successfully applied to the analysis of textual data for digital forensics.

1.5 Dissertation Outline

The remainder of this dissertation is organised as follows:

- **Chapter 2** examines the field of digital forensics, reviewing the current methodologies and approaches used to analyse digital data and to find and extract digital evidence. The need for tools that can quickly, efficiently and accurately analyse data to reveal truly useful information is highlighted.
- **Chapter 3** surveys the significant literature in text mining. Text mining techniques are explored and identified for use in digital forensics.
- **Chapter 4** presents a novel framework for evidence discovery in digital forensics.
- **Chapter 5** presents two models to find and extract relations between concepts from textual data, for performing the relation extraction process of the evidence discovery framework.
- **Chapter 6** discusses and compares the results of the two relation extraction models and evaluates the usability of the framework for evidence discovery.
- **Chapter 7** concludes this dissertation and discusses ideas for future work.

The appendices provide additional information that may assist the reader in understanding this dissertation:

- **Appendix A** lists and describes the part of speech tags used in this dissertation.
- **Appendix B** lists and describes the named entity tags used in this dissertation.
- **Appendix C** describes the force-based layout algorithm used to display the textgraphs created in this research.
- **Appendix D** presents the assignment of weights used in the evaluation.
- **Appendix E** provides a list of acronyms used in this dissertation.
- **Appendix F** lists and defines the mathematical symbols used in this work, categorised according to the relevant chapter in which they appear.

Additionally, an index is provided and begins on page 135 of the text.

Chapter 2

Introduction to Digital Forensics

This chapter reviews the current methodologies and approaches used to analyse digital data and to find and extract digital evidence. First an introduction to digital forensics is given, whereby digital evidence and digital forensic science are defined. Section 2.2 discusses the analysis phase of a digital investigation and highlights the challenges faced by investigators in the analysis of data to uncover potential evidence. The computer forensic software tools that are curretly available are then described and discussed in Section 2.2.1.

Section 2.3 surveys the current literature in digital forensic data analysis and summarises the advances made in the field. Section 2.3.1 focuses on the application of data mining techniques for use in digital forensics and approaches leading towards evidence discovery are discussed. The linguistic nature of digital evidence is then discussed in Section 2.3.2.

This chapter concludes by highlighting the requirements and needs for advancement in the analysis phase of a digital forensic investigation.

2.1 Introduction

The face of the world is changing. Our lives and social norms are shifting from reality to virtual reality. As everything from business to socialising and dating moves online, individuals are increasingly exposed to new and undesirable risks such as identity theft,

invasions of privacy, banking fraud, and the general illegitimate and unintended use of personal information. While simplifying and improving life and business in a myriad of ways, the virtual world simultaneously opens up a whole new dimension of criminal operation. Palmer (2001) speaks of a 'digital hand' that seems to have touched every corner of world culture. He predicted that, "because of this digital ubiquity, all crimes would soon have a *cyber-dimension*."

The extensive use of digital media such as email, documents, images, spreadsheets, the world wide web and networking, to name only a few, leaves a forensic trail of evidence. As Palmer predicted, the enormous archives of emails, business documents, and network logs hold the potential to serve as *cyberwitnesses* to *cybercrimes*. Digital forensics potentially holds the answers to the questions of who, what, why, where, when and how (Palmer, 2001). Palmer postulates that digital data might even provide clues about the motivation behind a crime or incident. Like a kitchen knife, the digital world can be used for good or bad ends and, whatever the use, internet users leave a trail of digital evidence behind them as they move through the virtual world. This dissertation uses the definition of digital evidence from Casey (2004):

> **Definition 2.1** (Digital evidence): any data stored or transmitted using a computer that support or refute a theory of how an offense occurred or that address critical elements of the offense such as intent or alibi.

What is important to note from Casey's definition, and even from Palmer, is that digital forensics is not limited to *cybercrimes* or crimes that were committed with the use of computers or in the virtual realm. Digital evidence can assist in an array of criminal and civil investigations ranging from fraud, identity theft, child pornography, homicides, kidnapping, abuse, and drug dealing.

Computer forensics played a pivotal role in identifying and eventually convicting the American serial killer, Dennis Rader, who murdered 10 people in Sedgwick County in Kansas (Bardsley *et al.*, 2005). Rader was particularly known for sending taunting letters to police and to local news papers boasting of the crimes and knowledge of details soon after each of the killings. His last letter, sent on February 16, 2005 was sent to FOX TV on a floppy disk. Using forensic software, EnCase (GuidanceSoftware, 2002), police

were able to examine metadata embedded in a deleted Microsoft Word document. This metadata revealed the name "Christ Lutheran Church" and showed that the document was last modified by "Dennis". The church's website published that Dennis Rader was president of the congregation council. Dennis Rader was arrested and later convicted (FOXNews, 2005).

In the case of Dennis Rader the digital data provided a link between the crime and its perpetrator; however, data does not have to provide a link to the perpetrator or victim to be useful in an investigation. Any number of pieces of useful information can be found in digital evidence, which can show how a crime was committed, provide investigative leads, disprove or support witness statements, and identify likely suspects (Casey, 2004).

Like the digital world, the field of digital forensics is still new. It is constantly developing and there is a strong need for standardisation on techniques and methods. The first Digital Forensic Research Workshop (DFRWS) was held in 2001 and since then, digital forensics has gained increasing attention. At that first DFRWS, the following definition for the science of digital forensics was constructed:

> **Definition 2.2** (Digital Forensic Science): The use of scientifically derived and proven methods towards the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorised actions shown to be disruptive to planned operations (Palmer, 2001).

As Palmer (2001) and Casey (2004) have pointed out, the investigation of almost any crime can potentially benefit from digital forensics, even those crimes which did not involve the use of computers or were not committed in the virtual world. It therefore follows that one is confronted with several different forms of digital evidence which might be usefully analysed: electronic media (examining physical media for evidence), software code (review of software for malicious signatures), and networks (scrutinise network traffic and logs to identify and locate suspicious activity and users) (Palmer, 2001). Each form of digital evidence has its own characteristics and requires different techniques to process and analyse the data.

Computer and network intruders, malicious hackers and computer software viruses have brought much attention to network and code analysis research. However, because of the amount of time it takes to analyse the data and the limited means of the techniques available, the use of media (computer hard drives and data) to support investigations has only recently picked up as a research area and has not been explored as much for assisting in the reconstruction of events and finding answers to crimes. For that reason, this research focuses on media analysis.

2.2 Analysis of Data to Uncover Digital Evidence

The obvious and immediate difficulty which confronts any person wishing to make use of the potential which digital evidence holds is that analysing and processing the ever-growing volumes of data, and linking the unstructured data together for use as evidence in a trial involves a tremendous amount of work. Examining and organising all of the data which digital evidence yields is a major challenge facing all investigators. For electronic evidence to become properly useful to investigators, accurate, efficient and rapid means of data analysis need to be developed (Chen *et al.*, 2004).

When a crime has been discovered that warrants the use of digital data collection, the first step is to collect and preserve all of the data in a legally sound way. In order for digital evidence to stand in a court of law, the investigator needs to be able to prove that this evidence has not been tampered with or manufactured. It is now widely accepted in courts to immediately preserve the data upon collection using specialised forensic software tools, such as EnCase (GuidanceSoftware, 2002). These tools are able to create a secure copy of the data in a non-invasive manner, whereby a physical bit-stream image of the subject's drive(s) is created. Such software tools also ensure that the data can be accessed and read, but not changed or overwritten in any way.

After the data has been preserved, these volumes must then be scrutinised and examined by an investigator. Similarly to when a physical crime scene is examined some evidence may be easier to find than others, for example, finding bullet casings may be easier to find than a relevant green thread caught on a window latch. In a digital crime scene the amount and type of data being examined influences the time and complexity

of the analysis process. Perhaps one could liken the examination of door entry and exit logs to establish the presence of a particular person to the searching for bullet casings. Consequently, the examination of business documents and emails to piece together a fraudulent transaction could be likened to finding that important green thread.

Palmer (2001) identifies three pertinent challenges facing digital investigators:

- Individual privacy and the collection and analysis needs of investigators continue to collide.
- Legal aspects need to be considered. If advanced technology is developed, but it does not comply with the law, the use of the technology is moot. Evidence laws need to be updated to accommodate digital evidence, and the investigation process has to follow legal processes, e.g. to assure that there is a clear chain of custody from the scene of the crime to the investigator and ultimately to the court. Only tools and methods that have been tested and evaluated to validate their accuracy and reliability should be used in practice.
- Tools are needed that zero in on truly useful information and quickly deduce whether it is material to the investigation or not.

This research will focus on improving the tools to assist with the process and leave the legal challenges for the legal experts. Computer forensic software exists that can assist in many phases of a digital investigation. However it is important to note that this software is not designed to solve crime. The primary goal of these systems is to reduce investigation time and complexity (Abraham, 2006). These tools focus on providing advanced search technologies with which an investigator can retrieve potential evidence. An investigator may use her/his experience and background knowledge of the case to select appropriate search terms and criteria to find clues in the data as to which data should be investigated in more detail or suggestions as to how to proceed with the investigation. For example, in the investigation of the disappearance of Chandra Levy, police discovered that she had visited a web site relating to Rock Creek Park on her computer the morning of the day she disappeared. Police searched this area and although they failed to find Levy's remains, which were nearby but outside the police's sweep, a

year later remains matching Chandra Levy's dental records were found in this area by a man walking his dog (Twomey and Horwitz, 2002).

Because of the enormous amount of time it takes to analyse data in this way a survey of the literature shows that the current tools available are not sufficient to meet Palmer's need to "zero in on truly useful information and quickly deduce whether it is material to the investigation or not." This research aims to explore the possibilities for advancing and improving these tools to more efficiently and effectively uncover digital evidence.

2.2.1 Computer Forensic Software Tools Currently Available

As mentioned above, in order to maintain the integrity of the data collected in an investigation, a secure image of the data needs to be created in a non-invasive manner using validated software for the digital evidence to be acceptable in a court of law.

A number of specialised tools are available to investigators for extraction, analysis and presentation of computer data in digital forensic cases such as: AccessData's Forensic Tool Kit (FTK) (AccessData, 2008), Guidance Software's EnCase Forensic Edition (GuidanceSoftware, 2002), Brian Carrier's The Sleuth Kit (TSK) (Carrier, 2008) and Dan Farmer and Wietse Venema's The Coroner's Toolkit (Farmer and Venema, 2007).

These tools also enable one to examine the image of the data without changing it in any way. The investigators' analysis may include many tasks such as examining file systems and meta-data structure details, reviewing the Windows registry for suspect information, looking up file hashes in a hash database, discovering and cracking passwords, reviewing lists of allocated and deleted ASCII and Unicode file names, and creating time lines of file activity. A typical forensic analysis will also include a manual review of material on the media, keyword searches for topics related to the crime, and extracting e-mail and images for review. Forensic software tools provide advanced regular expression and search syntax, comprehensive filter conditions and multiple sorting fields, which assist with searching for data and images. Thus an investigator may organize files based on their type (for example all executables, jpegs, and documents are separated) and quickly review numerous graphic images using pages of thumbnails. Such tools may also provide international language support and right-to-left reading, which searches for the keywords in a right-to-left sequence when needed.

2.3 Advances in Data Analysis

The difficulty with finding relevant information quickly using the current tools and methods is that they do not fully utilise the content of the data. While sorting files based on their type extension may be efficient to find and retrieve images, finding an essential word document may not be as easy. However, desktop search engines provide full text search of content and metadata on the user's PC, including web browser histories, e-mail archives, text documents, sound files, images and video, which greatly assists users to find data more quickly. IR applications have proven their success with their popularity and results that are so good these days that services offered by Google, Yahoo and Microsoft with their web search, desktop search and email search engines[1] are often taken for granted. The success of these search technologies makes them an obvious target for further research for their application in digital forensics (Beebe, 2007).

To improve evidence retrieval, one would typically aim to improve either the recall (the fraction of relevant hits retrieved out of all of the relevant hits, with respect to the query) or the precision (the fraction of relevant hits retrieved out of all of the retrieved hits, with respect to the query). Improving the recall may assist in finding latent evidence, but the information retrieved is still limited by the query terms used. Focusing on improving the precision reduces the amount of irrelevant information, thus making it easier and faster to find the critical information, but consequently increases the risk of losing potentially relevant information. According to Beebe (2007), investigators can find themselves wading through hundreds of thousands of search hits, of which estimates of 80-90% are irrelevant to investigative objectives. Thus even on a reasonably small device of 80GB, an investigator can become overwhelmed with a small set of queries and exhaust valuable resources and time scanning through the results.

Beebe (2007) propose that retrieved search hits should be clustered into similarly themed groups, which would enable the analyst to evaluate and discard irrelevant clusters and more quickly find relevant clusters. Only the clusters that are deemed relevant need to be closely scrutinised. Specifically, Beebe (2007) tests the practicability of using self-organising neural networks to thematically cluster text string search hits after the hits

[1]The word Google is now used more often as a verb than a proper noun.

have been retrieved.

However IR is only as good as the query terms used, which means that in a digital investigation the evidence retrieved is limited to the background knowledge of the case and extended search terms from the investigator's personal experience. Arguably the dependence on query terms and their many combinations makes searching for digital evidence very time consuming and inefficient. Since some evidence may be latent[2] due to poor file naming, organisation and searching, or purposefully concealed, this hidden data is very unlikely to be retrieved through traditional IR algorithms. In a perfect world, a forensic system could "discover" data that seems suspicious, whether latent or hidden. This has led researchers to investigate different approaches to finding evidence.

2.3.1 Towards Evidence Discovery

An interest has been sparked in the application of data mining and techniques for use in digital forensics (Chen *et al.*, 2004; Dozier and Jackson, 2005; Fan *et al.*, 2006; de Waal *et al.*, 2008) due to its promise to find new patterns and to discover knowledge that is held within vast amounts of data. Although it may be unrealistic at this point to aim for "automated evidence discovery," the techniques and algorithms from data mining, computational intelligence, linguistics, social network analysis, and visualisation may certainly make the analysis process faster, more efficient and more accurate (Palmer, 2001). Steps towards automated evidence discovery include profiling (Abraham, 2006), social network analysis (Chen *et al.*, 2004), and text mining (Dozier and Jackson, 2005; Fan *et al.*, 2006).

Investigative Profiling

Profiling is a technique often used in criminal investigations, where a profiler, based on her/his analysis of the crime data, predicts the type of personality which is likely to be involved in the perpetration of the crime in order to narrow the search for the

[2]Definition (latent): potentially existing but not presently evident or realized; "a latent fingerprint"; "latent talent" (wordnet, wordnet.princeton.edu/perl/webwn); existing or present but concealed or inactive (wiktionary, en.wiktionary.org/wiki/latent)

likely criminal. These profiles can then be used to identify suspects and rule out other potential suspects. Data mining techniques can be used to learn rules from data that characterises the behaviour of a computer user, much like a conventional profile. These profiles can then provide insight into the day-to-day use of the digital environment being investigated (Abraham, 2006). Banks have created profiles of credit card users to monitor their spending habits. A suspicious change in a user's behaviour (e.g. the purchase of an airplane ticket by someone who does not fly regularly) will then trigger an alarm for potential fraud, which the bank can then verify with the card holder to prevent the fraudulent use of the card. Fawcett and Provost (1997) investigated the application of such profiling to the problem of detecting cellular cloning fraud based on a database of call records. Using a large database of customer transactions, they use machine learning techniques to develop rules that characterise fraudulent behaviour. These rules, called indicators, are then used to monitor the profiles of customers, and to indicate anomalies and fraudulent transactions.

Abraham (2006) explored the analysis of profiles in two different ways: similarly to Fawcett and Provost (1997) a profile may be analysed on its own by looking into the contents of each profile to find discrepancies within the profile, or individual profiles may be grouped together based on their similarity and those profiles that lie outside major groups may be detected and investigated. Abraham (2006) created profiles based on the sequence of events extracted from data and used an event chain analysis approach to compare and analyse the profiles. This approach was tested on logged data from door swipe access in an office building. The algorithm was able to detect anomalies and unusual behaviour such as "shoulder surfing," where one person swipes their access card to open a door and the door is held open for a group of people to enter or leave.

Profiling also plays a large role in email analysis and can be used to characterise the author of emails. This may be used to identify the educational level or gender of an offender and may also establish if an email has been masqueraded (de Vel *et al.*, 2001). The propagation of emails can also be monitored to determine user cliques and behaviour profiles based on frequencies. These profiles can then be used to detect email virus propagation, spambot activity, and security policy violations (Stolfo *et al.*, 2006). Hershkop (2006) explores the use of profiles created from email data for a variety of

tasks, such as clustering and classification of similar emails, spam detection and forensic analysis to reveal information about user's behaviour.

Social Network Analysis

Similar to profiling, network analysis uses data mining techniques to characterise social and criminal networks, by describing the roles and interactions of the entities in the conceptual network (Abraham, 2006). Criminals often develop networks in which they form groups or teams to carry out various illegal activities (Chen *et al.*, 2004). Investigators can use social network analysis techniques to construct a network that illustrates criminals' roles, the flow of tangible and intangible goods and information, and associations among these entities. Criminal network analysis is a data mining task which consists of identifying subgroups and key members in such networks and then studying their interaction to develop effective strategies for disrupting the networks (Chen *et al.*, 2004). Chen et al. evaluated this method on data comprising of Tucson Police Department incident summaries, by extracting criminal relations based on weighted co-occurrences. Tucson Police domain experts confirmed that the sub-groups found, correctly represented the real group's organisation. Such criminal network analysis could greatly increase crime analysts' work productivity by discovering knowledge about criminal organisations that requires many hours to uncover manually.

Text mining

The adaptation and application of data mining methods to textual data is called text mining (Hearst, 2003). Text mining should ideally uncover something interesting about the relationships between text and the world, such as what actual persons or companies an article is discussing (Dozier and Jackson, 2005). Although text mining has not been explored very much, it is an important research direction that is now being explored by a number of researchers (Chen *et al.*, 2004; Dozier and Jackson, 2005; de Waal *et al.*, 2008; Fan *et al.*, 2006).

The automatic extraction of named entities (e.g. names of people, places, and companies) from textual documents, called named entity extraction, has shown great performance and success (Grishman and Sundheim, 1996). Named entity recognition was

investigated by Louis *et al.* (2006) for use in South African digital forensic investigations. Often, entities provide the unit of analysis in crime data mining applications. An important future research direction should investigate further extraction of information from textual documents such as the roles of entities and the relationships among them (Chen *et al.*, 2004).

References to professionals such as oncologists and cardiologists were extracted from newspapers, legal articles and medical articles by Dozier and Jackson (2005) as possible expert witnesses in their field who may have testified at a trial. Using text mining techniques, such as named entity recognition (NER) and information extraction (IE), Dozier and Jackson developed an application to automatically populate an expert witness database of medical practitioners.

Topic modelling captures the semantic context of a text collection and describes each topic by a set of words. De Waal *et al.* (2008) investigated the application of topic models on forensic data and its ability to contribute to digital analysis.

Text mining encompasses a number of information processing methods such as IE, topic tracking, content summarization, information visualisation, question answering, concept linkage, text categorization/classification, and text clustering (Fan *et al.*, 2006). Some of these research areas are described in more detail in the following chapter.

Information Visualisation

The visualisation of data can be more than just the presentation of data analysis results. In fact information visualisation can be a data analysis approach on its own (Hotho *et al.*, 2005). The human mind has amazing pattern recognition capabilities that far surpass those of the most advanced computational intelligence algorithms. According to the developers of Starlight (Woodson, 2006), the key to harnessing the mind's visualisation power for problem solving is to capture information relationships in meaningful ways, and to present these relationships in intuitive graphical forms (Woodson, 2006). By transforming textual data into a graphical form such as hierarchical concept maps, timeline representations, graphs and social networks, the underlying patterns in the data may be exposed and identified by the investigator.

Starlight (Woodson, 2006) is a proprietary visualisation tool for general purpose in-

formation presentation, information modelling and data management. It has a powerful graphical interface and has a variety of 3D forms for visualising information. This visualisation platform demonstrates the advantages and applicability of the visualisation of information for data analysis. Currently, the use of visualisation in forensic software, if used at all, has been limited to the simplistic presentation of search results. To overcome this limitation, Vlastos and Patel (2008) developed a visualisation tool for specific application in cyber forensics. They use a variety of colours and 3D block drawings to depict the file system and provide an intuitive view of deleted, wiped, encrypted and transformed files. A user interface enables an investigator to navigate through the file system and explore the file list in a way that is optimised for presenting evidence in digital investigations.

However, the power of visualisation that has been used so successfully in many scientific fields has not yet been exploited for cyber forensics. Visualisation shows great potential for explorative evidence discovery and assisting with data analysis. If the need for tools that quickly find important information and assist an investigator to quickly decide whether that information is relevant is to be met, then the visualisation channel of the human mind is a resource that undoubtedly needs to be tapped into.

2.3.2 Linguistic Nature of Digital Data

Before methods to assist with evidence discovery on digital media can be developed, the type of data that is prevalent must be considered. Common estimates state that over 80% of business information and information in general is stored in a textual format (TextAnalytics, 2005). Consider the types of files that are worked with, accessed or inadvertently modified on a daily basis: word processing documents, spreadsheets, presentations, emails, address books, calendar appointments, internet browsing (history logs and content), instant messaging, network activity logs, and system logs. Although these data types all differ in format, these data types are all partially or completely textual and linguistic in nature, i.e. the information is expressed in human languages, programming languages, or system and application logging conventions (Beebe, 2007). The expressivity of language makes these files rich information sources, which makes textual data and consequently textual evidence very important in digital investigations.

Text and especially human or natural language is very complex and difficult to work with. Although it is quite easy for a human to understand and process natural language, it is very difficult to build a computer system that can interpret the ambiguities and semantics behind the symbols and syntax of language. There has been great success in the field of natural language processing in developing algorithms to process syntactic structures, such as part-of-speech tagging, phrase chunking and full parsing. However, there still exists a barrier preventing computational algorithms from revealing the semantic meaning of human language.

Structuring the textual data applies labels to pieces of the text thus providing access to some of the semantics. For example, an email has both semi-structured and unstructured parts. The header contains fields for the sender's name, the sender's email address, the recipient's name, the recipient's email address and a subject. It is therefore a simple algorithmic problem to extract the names of the sender and recipient. The subject line may provide a summary of what the email contains, but gaining a semantic understanding may still be difficult. The majority of the information contained in an email is usually in the unstructured parts of the body and any attachments. The body of the email is written natural language, albeit sometimes poorly written, which can be a challenge on its own. Access to textual data is currently dominated by search based methods, and due to the complexity of natural language, it is very difficult to develop more advanced algorithms.

Typical data analysis techniques and data mining methods, such as association analysis, classification and prediction, cluster analysis, and outlier analysis identify patterns in structured data (Frawley *et al.*, 1992; Hotho *et al.*, 2005). For example, association mining is used to analyse consumer purchases to develop recommender systems, such as that used by Amazon (Agrawal and Srikant, 1994). If such methods can be adapted and extended to unstructured data, the wealth of information stored in textual data may be accessed more quickly (Hearst, 2003). A move towards this goal has taken place with text mining methods such as content summarisation, document classification and clustering techniques (Hotho *et al.*, 2005).

2.4 Conclusion

The use of media analysis to assist with criminal investigations is currently limited by the amount of time it takes to analyse the volumes of data and the limited means of the techniques available. Examining and organising the data to piece together the events and facts of a crime is a major challenge for all investigators who wish to utilise digital evidence. A review of the literature shows a general call for the extension of text mining research to assist in answering the need for tools that can quickly, efficiently and accurately analyse data to reveal truly useful information (Palmer, 2001; Beebe, 2007).

The difficulty with finding relevant information quickly using the current tools and methods is that these tools rely very heavily on background knowledge for query terms and do not fully utilise the content of the data. A move away from IR techniques includes text summarisation and document classification or clustering. These techniques reduce the amount of data being scrutinised. However, a deep examination of the data is still required to zero in on the relevant information or evidence. To make the investigation process more efficient, tools need to be developed that will reduce the quantity of data, aid the analysts' exploration of the data, and enhance the intelligibility of the presentation of the data. This research will investigate the extension of text mining research for digital forensics in an attempt to meet these criteria. The following chapter presents a survey of the literature on text mining.

Chapter 3

Survey of Text Mining Methods

This chapter presents a survey of the literature on text mining. The introduction defines what text mining is and dicusses its origin and what it entails. The following sections highlight the prominent research in each of the main analysis tasks in text mining, namely text pre-processing (Section 3.2), classification and clustering (Section 3.3), association mining (Section 3.4), information extraction (Section 3.5) and visualisation for explorative text mining (Section 3.6).

This chapter concludes by presenting a brief summary of the text mining methods disussed and highlights the limitations of text mining.

3.1 Introduction

There is some confusion as to what text mining is. Text mining is often thought of as knowledge discovery from text, an extension of knowledge discovery from databases (KDD). Knowledge discovery is a life cycle of a series of partial steps, following the CRISP-DM process model (Chapman *et al.*, 2008). To understand what text mining is, what techniques have been developed, what it aims to accomplish, and where it originated from, its limitations and the data that it is intended to work with should be considered.

The idea behind text mining, or text data mining, is to extract patterns from large amounts of textual data. Text mining stems from the field of data mining. Data mining

aims to uncover interesting patterns or trends from data in large databases. Frawley *et al.* (1992) define data mining as "the nontrivial extraction of implicit, previously unknown, and potentially useful information from data." For example, data mining techniques are used to learn which products consumers typically purchase together to assist with marketing and sales. Thus, if analysts identify that consumers who typically buy baking trays also buy a recipe book with it, these items can be placed in close proximity to each other in the store or marketed together as part of a special promotion. This technique forms the basis of recommender systems often used on web based stores, such as Amazon.

Data mining is designed for use on databases. Databases are designed to enable computers to easily access and automatically process data contained in the database. Data mining is therefore designed for use on text which is constituted completely differently to the text which forms the input for text mining. Textual data for text mining is in the form of natural language, which is meant for humans to read and understand. Natural language is unstructured while the algorithms designed for data mining require structured, computer data. To a limited extent, algorithms can identify and interpret the syntactic structure of text, or classify or cluster words and documents based on statistical measures. However, it is questionable whether a computer program could ever match the ability of the human mind in its ability to read and interpret natural language and its context? Hearst (2003) believes that it may require a full simulation of how the mind works before programs can be written to read the way that humans do.

In its simplest terms, what the human mind appears to do is to recognise and process patterns in complex text. Algorithms that tackle tasks relating to the modelling of hidden patterns in text and perform text analysis have been produced in related research areas such as IR, question answering, computational linguistics, natural language processing (NLP), and text summarisation (Jurafsky and Martin, 2008). The low-level algorithms or subtasks in these related research areas have grown towards the high-level goal of IE and text mining (Hearst, 2003). These small sub-tasks may bridge the gap between mining structured data and unstructured text. The experience and results from these areas influence the on going research in text mining.

As described above, these subtask algorithms have emerged from the fields of IR, question answering, computational linguistics, NLP, and text summarisation. IR is the

process of finding documents which contain answers to questions. IR does not retrieve the actual answer. Search engines like Yahoo! retrieve documents based on keywords, known as document retrieval, but are frequently referred to as IR systems (Hotho *et al.*, 2005). As the description of IR suggests, IR is closely related to research on question answering, which aims to retrieve the answer itself.

The fields of NLP and computational linguistics both aim to achieve a better understanding of natural language through automated computational processes (Kodratoff, 1999; Jurafsky and Martin, 2008). Processes that perform 'natural-language-understanding' convert samples of human language into more formal representations that are easier for computer programs to manipulate. Processes that perform 'natural-language-generation' convert information from computer databases into normal-sounding human language.

Automatic text summarisation aims to create a shortened or condensed form of a document that contains the key aspects of the original document. Standard approaches for automatic text sumarisation are based on the principle of selecting and extracting pertinent sentences that contain the most important information. The summary is then made up from the collection of these sentences (Jurafsky and Martin, 2008). However, some attempts have been made to extract semantic information from documents and create summaries based on this information (Leskovec and Grobelnik, 2004; Hachey and Grover, 2005).

As Hotho *et al.* (2005) have highlighted, the definition of text mining depends on the research area and focus. Text mining can mean information extraction, or text data mining, or it may be defined in the broader sense of knowledge discovery as in the KDD process. This dissertation focuses on methods that extract useful patterns from text in order to categorise or structure text collections, and methods to extract useful information, following the definition chosen by Hotho *et al.* (2005) of text mining as text data mining. The rest of this chapter covers the main analysis tasks in text mining, namely pre-processing, classification and clustering, association mining, information extraction, and visualisation for explorative text mining.

3.2 Text Pre-Processing

Computers handle text as simple sequences of character strings. Therefore pre-processing algorithms are used to extract and store the information from text documents in a data structure that is more appropriate for further processing than a plain text file (Hotho *et al.*, 2005).

There are a number of different types of pre-processing that can be done, and the appropriate one should be chosen depending on the text mining algorithm which is intended to be used. This section covers the main pre-processing algorithms available, which are used in the common text mining methods discussed in later sections of this dissertation.

The most common approach to pre-processing is what is known as a 'bag of words' approach. This is based on the idea that a text document can be represented by a collection of words. The importance of words within the given document can be defined by assigning the word a numerical value (Hotho *et al.*, 2005) using a vector space representation such as the vector space model (Salton *et al.*, 1975), the probabilistic model (Robertson, 1977), and the logical model (Rijsbergen, 1986).

An alternative to the 'bag of words' approach is linguistic pre-processing. Liguistic pre-processing exploits the syntactic structure and semantics of the text in various ways, depending on the task, and could either assign syntactic labels to words, store words in pairs or triples, or in the form of a dependency tree.

Tokenisation is the process of recognising the basic units within the text by segmenting the stream of characters into separate parts or tokens, in order to obtain all the words, punctuation, numerical, monetary, time, date and other expressions in the text (Grefenstette and Tapanainen, 1994). Simply, tokenisation splits text into sentences and sentences into words.

Filtering aims to remove words which contain little semantic information and are considered to be noisy (e.g. the definite and indefinite articles 'the' and 'a', conjunctions and prepositions), from the text. These words are referred to as stop words and there is no definite list as to which words should or should not be removed. Filtering helps with statistical methods that use the frequency of a word occurring within a document to characterise and distinguish that document from other documents. Thus words that

occur either extremely frequently or very seldom within a corpus may be considered to be of little statistical relevance and may also be filtered out.

Lemmatisation is the process of grouping together the different inflected forms of a word so they can be analysed as a single item (Brown, 1993). For example, the verb 'to walk' may appear as 'walk', 'walked', 'walks', 'walking'. The base form, 'walk', is called the *lemma* of the word. The process involves tagging each word with its part-of-speech tag and then applying normalisation rules to map verb forms to their infinite tense and nouns to the singular form. Lemmatisation algorithms are time consuming, error-prone and difficult, and stemming methods are therefore often preferred (Hotho *et al.*, 2005).

Stemming algorithms reduce inflected words to their stem or base form, which need not be identical to the *lemma* of the word. There are a number of different approaches to stemming such as 'brute force' algorithms, suffix stripping algorithms, and stochastic algorithms. These methods aim to strip the prefixes, suffixes and other affixes from words to form a natural group of related words with equal or similar meaning. For example, the 'ed' is removed from 'walked', the 'ing' is removed from 'walking' and the 's' is removed from plural nouns.

Indexing or keyword selection is used to reduce the dimensionality of the data by selecting keywords that represent the document, rather than using all of the words contained within the document. A simple method for keyword selection is based on the entropy of the words in the document. In this context, the entropy is a measure of the importance of a word in a given domain, such that words that occur frequently in many different documents will have low entropy and words that are used in a specific domain and in a limited context will have a high entropy or information value. Thus entropy can be used as a measure of how well a word can be used to differentiate between documents in a corpus, and words with high entropy can be selected as keywords for a document. Lochbaum and Streeter (1989) define the entropy of a word $t \epsilon T$ as:

$$W(t) = 1 + \frac{1}{\log_2 |D|} \sum_{d \epsilon D} P(d,t) \log_2 P(d,t) \quad \text{where} \quad P(d,t) = \frac{t_{freq}(d,t)}{\sum_{l=1}^{n} t_{freq}(d_l,t)} \tag{3.1}$$

where D is the set of all documents and $T = \{t_1, \ldots, t_m\}$ represents all the words in the dictionary, i.e. the set of all different terms occurring in D. The absolute frequency of

term $t \epsilon T$ in document $d \epsilon D$ is given by $t_{freq}(d, t)$.

The *term frequency — inverse document frequency (tf.idf)* weight is an alternative statistical measure used to evaluate how important a word is to a document (Salton and McGill, 1983). Similarly to the entropy discussed above, the importance of a word increases proportionally to the number of times that word occurs in the document, but is penalised by the frequency with which the word appears in the corpus. The term frequency of a word or term is given by $t_{freq}(d, t)$, as above. The inverse document frequency (idf) is a measure of the general importance of a word. The idf is calculated by dividing the number of all documents, $|D|$, by the number of documents containing the relevant word or term, $df(t)$, and then taking the logarithm of the quotient as follows:

$$idf(t) = \log \frac{|D|}{df(t)}. \tag{3.2}$$

Thus the idf of a rare term is high, whereas the idf of a term that occurs in every document will be zero. The tf.idf weight is calculated by:

$$tf.idf = t_{freq}(d, t) * idf. \tag{3.3}$$

The weight assigned by the tf.idf will be highest when the word or term occurs many times within a small number of documents and will be lower when the term occurs fewer times in a document, or occurs in many documents (Manning *et al.*, 2008).

An alternative and often preferred method for keyword selection is the use of latent semantic indexing (LSI) (Deerwester *et al.*, 1990). LSI manipulates the term-document association data to reveal an underlying semantic structure by allowing this structure to reflect the major associations within the data. This compensates for the variety of words that people use to describe the same thing, *synonymy*, and the use of the same word with differing meanings, *polysemy*. As a result, terms may be selected for indexing a document if the terms are close to the document in the new 'semantic' structure even if they do not actually appear in that document.

LSI is achieved through the use of singular-value decomposition, which is closely related to eigenvector decomposition and principal component analysis for data dimensionality reduction. This method is based on a word-document association matrix which is decomposed into a set of orthogonal factors from which the original matrix can be

approximated by linear combination. The idea is that, by retaining only the largest singular values of the diagonal matrix along with their corresponding columns in the other matrices, the major associational structure is captured and the noise is discarded, resulting in a new smaller matrix (Dumais *et al.*, 1988).

Linguistic pre-processing incorporates a number of methods and algorithms, most of which originated in the field of natural language processing (Manning and Schütze, 2001; Jurafsky and Martin, 2008). These methods extract or label additional information about the words or text to reveal information about the syntactic structure, such as:

- Part of speech tagging determines the part-of-speech of each word in a sentence based on its definition and context, and applies a label or tag (e.g. noun, verb, adjective, etc.) to each word. While there are a number of both stochastic and rule based approaches, the most well known algorithm is the Eric Brill tagger (Brill, 1992).

- Phrase recognition, known as chunking, segments sentences into their subcomponents by grouping or 'chunking' adjacent words into phrases. For example, the chunks "the tile rooftop" and "especially slowly" would be recognised as noun and adverbial phrases respectively.

- Word sense disambiguation, which is the task of discerning the meaning of a word in a given context to resolve the ambiguity of which sense of the word is intended. For example, 'bank' in the noun form may refer to the 'financial institution' or the 'border of a river', or in the verb form it may mean 'to cash in', or 'to tilt'.

- Parsing, which assigns the relation of each word in a sentence to each other word and represents the sentence in a tree structure. Each word is annotated with its grammatical type and its function in the sentence, e.g. subject, object, etc. (Jurafsky and Martin, 2008).

3.3 Classification and Clustering

Very large document collections are difficult to manage and store in a manner that makes it easy to find relevant information at a later stage. Typically, these collections

would be indexed or catalogued in a library, archive, or repository. However, manually maintaining such a system is very time consuming making it difficult to keep up to date with the indexing and cataloguing of the new documents which are continually being generated. Automated document collection structuring techniques have been developed to categorise or group the documents as ways to relieve this problem.

Classification or categorisation methods (Cavnar and Trenkle, 1994; Dörre *et al.*, 1999; Lodhi *et al.*, 2002; Hotho *et al.*, 2005) use a pre-defined taxonomy and assign classes or keywords to new documents. The classes or categories in the taxonomy are designed for the expected domain and the intended use of the document collection.

Clustering techniques (Dörre *et al.*, 1999; Steinbach *et al.*, 2000; Hotho *et al.*, 2005) are sometimes preferred as they are fully automated and do not depend on a pre-defined taxonomy. Clustering techniques group documents into unlabelled classes based on similarities between documents. Terms can then be extracted to describe or label these classes.

Although classification and clustering methods were originally designed to assist with document search and IR, they can also greatly assist with finding similar documents (nearest neighbours) and with improved understanding, browsing, and summarisation of a document collection.

This sections describes these two types of methods. Classification methods are discussed in Section 3.3.1 and Clustering methods are discussed in Section 3.3.2.

3.3.1 Classification

As stated above, text classification is the process of assigning labels or classes to new documents to be added to a collection. Such a system could be used to route electronic news articles into categories such as “finance”, “politics” or “sport”, or to assist with labelling and sorting documents based on the language in which they are written, e.g. French, German and Italian, amongst others.

Classification systems require a pre-defined taxonomy of classes and a set of documents that have already been labelled and assigned a class. These labelled documents are then used to develop and train a model, which should then be able to correctly assign a class, or classes, to a new document in the domain. Typically, a bag-of-words

approach is used. However, a number of different design choices can be taken for the task of classification. Cavnar and Trenkle (1994) used an n-gram approach with frequency profiles, while Lodhi *et al.* (2002) preferred the use of string kernels and a support vector machine. Sebastiani (1999) and Hotho *et al.* (2005) provide an overview of a range of standard approaches.

The definition of the classes (sometimes called the taxonomy), and labelling of the set of training documents is typically prepared manually. However Dörre *et al.* (1999) automated this labelling process to some degree by providing a set of sample documents for each class and then extracting keywords to characterise each class, a technique which is very similar to clustering.

The performance of a classification system is typically measured against a small subset of the original training data that is held back for testing purposes. This portion of documents can then be classified using the model, and the new estimated labels are compared to those that were manually assigned. The *accuracy* of the system is measured as the fraction of correctly classified documents out of the total number of documents to be classified. However, it is common that a certain class may dominate, and a very large percentage of all the documents may belong to that class. Thus a high accuracy may be reached by assigning all documents to the dominant class (Hotho *et al.*, 2005). To compensate for the possibility of skewed *accuracy*, it is standard practice to also calculate the *precision, recall* and *F-score*. Precision calculates the fraction of labels that have been correctly assigned out of all of the assigned labels, i.e. of all the labels which were assigned, how many were correct. Recall calculates the fraction of labels that were correctly assigned out of all the relevant documents that should have that label, i.e. of all the documents that should be in a class, how many are assigned to that class. Precision and recall are defined with respect to IR in terms of a set of retrieved documents and a set of relevant documents, and are defined as:

$$precision = \frac{\#\{relevant \cap retrieved\}}{\#retrieved} \qquad recall = \frac{\#\{relevant \cap retrieved\}}{\#relevant} \tag{3.4}$$

The F-score calculates the harmonic mean of precision and recall and gives an overall measure of the performance. This enables the best trade-off between precision and recall to be established. The F-score is defined as:

$$F = \frac{2 * precision * recall}{(precision + recall)} \tag{3.5}$$

3.3.2 Clustering

Clustering is very similar to classification in that it aims to assign documents into related groups or 'unlabeled classes' that have similar content. The advantage of clustering is that it is fully automated and does not require a set of labelled documents to develop the model. These techniques partition the document collection into groups based on their content or features that have been extracted from the documents. Clustering algorithms can either partition the collection into a flat structure or an hierarchical structure of groups (Steinbach *et al.*, 2000). A list of keywords or terms that are common to the documents in a group may be used to describe the group and in doing so provides an overview of the contents of the collection (Dörre *et al.*, 1999).

The performanc of clustering algorithms is typically quantified using statistical measures of similarity and dissimilarity. Although the ideal clustering result is different depending on the application, the data, and the user, the quality of clustering is considered to be better if documents within a cluster are very similar to each other and more dissimilar between clusters (Hotho *et al.*, 2005). There are numerous clustering algorithms and evaluation methods for these algorithms as overviewed by Steinbach *et al.* (2003).

3.4 Association Mining

The use of association rule mining has had great success in the field of data mining for discovering interesting relations between variables in large databases (Agrawal and Srikant, 1994; Feldman and Hirsh, 1996; Bayardo, 1998; Lin and Kedem, 1998; Zaki, 2000; Han *et al.*, 2004). This can be seen in particular in the application of market basket analysis, whereby transactions recorded by point of sale systems are analysed to identify trends or patterns, called association rules (Agrawal and Srikant, 1994). A typical example of such a rule found by association mining might be that 83% of customers who purchase milk and bread also buy eggs. Identifying such rules can be used for a

variety of marketing tasks such as promotions, store layout, and catalogue design. Apart from market basket analysis, association mining has been used in a variety of fields from bioinformatics and intrusion detection to web usage mining (Agrawal and Srikant, 1994; Feldman and Hirsh, 1996; Bayardo, 1998; Lin and Kedem, 1998; Zaki, 2000; Han *et al.*, 2004).

The problem of association rule mining was formally stated in the work of Agrawal *et al.* (1993). An *association rule* is an implication of the form $X \Rightarrow Y$, where $X \subseteq I, Y \subseteq I$, and $X \cap Y = \emptyset$ and $I = i_1, i_2, \ldots, i_m$ is the set of all items. The rule $X \Rightarrow Y$ has *support*, s, in the transaction set if $s\%$ of transactions contain $X \cup Y$. The rule $X \Rightarrow Y$ holds in the transaction set with a confidence c, where c is defined as $confidence(X \Rightarrow Y) = \frac{support(X \cup Y)}{support(X)}$.

Finding all association rules consists of two steps: 1) finding all sets of frequently co-occuring items called frequent sets or large item-sets and 2) using the frequent sets to generate all the association rules. Many algorithms have been developed for fast association mining, with perhaps the most well known being the Apriori algorithm (Agrawal and Srikant, 1994). Other algorithms which have been developed include MaxMiner (Bayardo, 1998), Pincer-Search (Lin and Kedem, 1998), Eclat (Zaki, 2000), and FP-Growth (Han *et al.*, 2004).

Due to its success, many people have tried to adapt association mining to applications in the textual domain using unstructured textual data (Raghavan and Tsaparas, 2002; Feldman and Hirsh, 1996; Rajman and Besancon, 1998). Association mining has been applied to text to extract, group, and organise the concepts that recur in a corpus of documents (Raghavan and Tsaparas, 2002). Varying approaches to find the key themes in a corpus have been taken. Feldman and Hirsh (1996) utilised a query-centred view to discover associations amongst keywords. Rajman and Besancon (1998) exploited the use of association rule mining in a "full text" approach by developing prototypical documents, being a document representing a class of similar documents in the corpus. Typically, performing association rule mining with textual data requires the use of common statistical measures such as document frequency, inverse document frequency, sentence frequency, and inverse sentence frequency, however additional NLP has occasionally been included.

3.5 Information Extraction

The goal of IE methods is to automatically extract useful information, patterns or trends from natural language text collections. These techniques can perform generic term or topic extraction (Hotho *et al.*, 2005) or rather selective extraction of specified types of information of particular semantic classes of objects (entities), relationships among these entities, and events in which these entities participate (Yangarber *et al.*, 2000).

IE tasks usually consist of a series of processing sub-tasks including tokenisation, sentence segmentation, part-of-speech assignment, and the identification of entities and events (Hotho *et al.*, 2005). The required pieces of information are often extracted and stored in tables or database-like patterns which are then available for further processing or use (Sekine, 2006).

This section describes the main tasks which fall under IE namely: term extraction, topic, trend and theme extraction, named entity extraction, and event extraction. The main information extraction pattern models are then described.

3.5.1 Term Extraction

Term extraction is the process of selecting concepts or words that co-occur frequently within a specific domain. Term extraction is used for a variety of tasks including feature extraction and data preparation for text mining (Carenini *et al.*, 2005; Azé *et al.*, 2005), terminology extraction for specific domains (Daille, 1996), or to discover thematic words or themes from a document collection (Mei and Zhai, 2005). Feldman *et al.* (1998) found that performing text mining at the higher term level rather than at the word or n-gram level achieved more efficient and more understandable results. Statistical measures can be used to determine the strength of the association between words. The measure used should be chosen so as to most suit the intended corpus and goals (Azé *et al.*, 2005).

Most term extraction methods use a combination of co-occurrence frequency and at least one other statistical measure most suited for the applicable corpus. Feldman *et al.* (1998) and Daille (1996) focus on the measures for co-occurrence frequency, association ratio, Φ^2, and log-likelihood. Co-occurrence frequency is the simplest measure and is determined by the number of times two words occur together.

Church and Hanks (1990) introduced the *association ratio* for measuring word association norms based on the concept of *mutual information*. Church and Hanks (1990) state that if two words x and y have probabilities of occuring, $P(x)$ and $P(y)$, their mutual information, $MI(x, y)$, is defined as:

$$MI(x, y) \equiv \log_2 \frac{P(x, y)}{P(x)P(y)} \tag{3.6}$$

The probabilities $P(x)$ and $P(y)$ are estimated by $f(x)$ and $f(y)$, the number of occurrences of x and y in the text, normalised by the size of the text. The association ratio measures the probability of the words x and y occurring together (the joint probability) with the probabilities of the words x and y occurring independently. The joint probability $P(x, y)$ is estimated by $f(x, y)$, the number of occurrences of x and y together. Thus if there is a genuine association between x and y, then the joint probability, $P(x, y)$ will be much larger than chance, $P(x)P(y)$, and consequently, $MI(x, y) \gg 0$. However, when $MI(x, y) \approx 0$ the probability that the two words occur together by chance is almost equal to the probability that their co-occurrence is due to a genuine association (Church and Hanks, 1990).

Although the association ratio is based on the information theoretic concept of *mutual information*, it is different from that concept in that joint probabilities are supposed to be symmetric. That is, $P(x, y) = P(y, x)$. Therefore, mutual information is symmetric, i.e. $MI(x, y) = I(y, x)$. However, the association ratio is not symmetric, since $f(x, y)$ encodes linear precedence, where $f(x, y)$ denotes the number of times x appears before y (Church and Hanks, 1990).

Gale and Church (1991) proposed a χ^2-like statistic coefficient, referred to as Φ^2. This coefficient is bounded between 0 and 1, whereby a contingency table can be computed for the association of two words x and y from $f(x)$, $f(y)$ and $f(x, y)$, and the size of the dictionary, N. Then,

$$\Phi^2 = \frac{(ad - bc)^2}{(a + b)(a + c)(b + c)(b + d)} \tag{3.7}$$

where,

$$a = f(x, y) \tag{3.8}$$
$$b = f(x) - f(x, y) \tag{3.9}$$
$$c = f(y) - f(x, y) \tag{3.10}$$
$$d = N - a - b - c \tag{3.11}$$

The log-likelihood ratio, LogL, introduced by Dunning (1993), presents a measure that does not depend on assumptions of normality, but rather uses the distribution of the generalized likelihood ratio. The practical effect of this improvement is that statistical textual analysis can be done effectively with significantly smaller volumes of text than is required for conventional tests based on assumed normal distributions. The log-likelihood ratio can be presented in terms of the contingency table described above as:

$$\begin{aligned} LogL &= a\log a + b\log b + c\log c + d\log d \\ &- (a+b)\log(a+b) - (a+c)\log(a+c) \\ &- (b+d)\log(b+d) - (c+d)\log(c+d) \\ &+ (a+b+c+d)\log(a+b+c+d) \end{aligned} \tag{3.12}$$

Additional measures of association to those already mentioned were considered by Azé *et al.* (2005), such as the mutual information with cube measure (Daille *et al.*, 1998), dice coefficient (Smadja *et al.*, 1996), number of occurrences with log-likelihood (Roche *et al.*, 2004), and the J-measure (Smyth and Goodman, 1992).

3.5.2 Topic, Trend and Theme Extraction

To gain a broader sense of the semantic contents of a document or document collections, term extraction has been extended to detect and extract topics or themes (Rajaraman and Tan, 2001; Zhai *et al.*, 2004). A topic can be modelled by a probabilistic distribution of words that describes and characterises a semantically coherent event or subject to reveal the underlying semantic context of a document collection. Extracted topics can suggest prevalent themes, which can be analysed or tracked over time. Topic detection and tracking (TDT) is a (DARPA) sponsored effort (Rajaraman and Tan, 2001) that

provides a framework and evaluation program for analysing trends from time-ordered information sources, e.g. electronic news.

Topics or themes can be extracted using a variety of models or methodologies. Zhai *et al.* (2004) proposed a generative probabilistic mixture model to discover common themes across documents to perform comparative text mining. Comparative text mining summarises the similarities and differences between the collections along each common theme.

Mei and Zhai (2005) also used a simple probabilistic model to extract themes, whereby words are regarded as data drawn from a mixture model with component models for the theme word distribution. By extracting and tracking themes, interesting temporal patterns in a collection of news articles and research papers can be identified, using the publication dates as timestamps.

Rajaraman and Tan (2001) proposed a clustering approach using adaptive resonance theory (ART) networks, a class of self-organising neural networks, to analyse trends from a stream of text documents. Topics (represented by clusters) are identified by the system and tracked by associating new incoming stories with the discovered topics.

3.5.3 Named Entity Extraction

Research in the related areas of question answering, IE, and NLP has identified the need for and importance of recognising information units. These information units include person, organisation and location names, and numerical expressions including time, date, money and percentage expressions (Nadeau *et al.*, 2007). The task of extracting these units was formulated in the 6th Message Understanding Conference (Grishman and Sundheim, 1996) and the term "named entity (NE)" was given to these units by Grishman and Sundheim (1996). Extracted NEs are often used as the basic unit from which further analysis can be performed in text mining applications (Chen *et al.*, 2004).

The field of named entity recognition (NER) and classification has developed significantly and the progress and current gold standard approaches are well documented in the survey by Nadeau *et al.* (2007). Research and evaluation programs for this task now cover a wide spread of languages with support for approximately twenty languages (Nadeau *et al.*, 2007), including German (Daelemans and Osborne, 2003), Spanish and

Dutch (Roth and van den Bosch, 2002), and Japanese (Uchimoto *et al.*, 2003). However, according to a survey by Nadeau *et al.* (2007), most of the work has concentrated on limited domains and textual genres such as news articles and web pages.

The task of extracting NEs can be seen as a classification problem or word-based tagging problem. Initial methods for NER were rule based. However, a move towards supervised machine learning algorithms has emerged as the current preferred approach (Hotho *et al.*, 2005). While supervised learning systems have a lower system engineering cost, they require a large collection of annotated data. Such collections are available from the evaluation forums (MUC (Muc, 1998), IREX (Sekine and Eriguchi, 2000), CONLL (Daelemans and Osborne, 2003) and ACE (Doddington *et al.*, 2004)) but remain rather rare and are limited in domain and language coverage (Nadeau *et al.*, 2007). The main techniques currently utilised for NE extraction include hidden markov models (Bikel *et al.*, 1999), support vector machines (Takeuchi and Collier, 2002; Asahara and Matsumoto, 2003), maximum entropy models (Borthwick *et al.*, 1998), boosting and voted perception (Collins, 2001), and conditional random fields (McCallum and Li, 2003).

3.5.4 Event Extraction

Event extraction, usually simply referred to as information extraction, is concerned with extracting particular pieces of information about a particular type of news event that has been reported again and again over time in news articles (Shinyama and Sekine, 2006). This task was initiated through the Message Understanding Conference (MUC) challenges, each of which focused on extracting information about events for specific domains, such as terrorism (Sundheim, 1992; Riloff, 1996), management succession (Sundheim, 1995; Yangarber *et al.*, 2000), or job announcements (Califf and Mooney, 2003; Freitag and McCallum, 2000). The required pieces of information are typically extracted to fit templates regarding *who* did *what* to *whom, when* and *where*, and eventually *why*, and stored in tables or database-like entries (Surdeanu and Harabagiu, 2002).

The following example shows the description for an event from the management succession scenario from the MUC6 challenge (Sundheim, 1995):

"Yesterday, McCann made official what had been widely anticipated: Mr.

James 57 years old is stepping down as chief executive officer on July 1 to retire. He will be succeeded by Mr. Dooner, 45."

The task is then to extract the organisation involved (McCann), the job position (chief executive officer), the date of the succession (July 1), the name of the predecessor or out-going person (Mr. James), and the name of the new appointee or incoming person (Mr. Dooner).

Most IE systems are developed for a specific domain where the kind of semantic information that is required is determined in advance. These systems usually use extraction patterns or rules to identify event descriptions and use annotated training data to learn pattern matching rules based on lexical, syntactic, and/or semantic information (Patwardhan and Riloff, 2006; Shinyama and Sekine, 2006; Hotho *et al.*, 2005). Thus extraction normally consists of an analysis of the text in terms of general linguistic structures and domain-specific constructs, followed by a search for scenario specific patterns (Yangarber *et al.*, 2000).

However, the flexibility and variety of natural language makes it very hard to find all forms of event descriptions and each mention of events, and requires systems to analyse very large corpora of text in the specific domain (Patwardhan and Riloff, 2006). This has lead a number of researchers (Yangarber *et al.*, 2000; Surdeanu and Harabagiu, 2002; Hasegawa *et al.*, 2004; Shinyama and Sekine, 2006; Sekine, 2006) to recognise the need to use unannotated data to improve the portability and performance of these systems.

Yangarber *et al.* (2000) proposed an approach to IE which uses a small set of 'seed patterns' to identify a set of relevant documents and a set of event patterns from unannotated text to perform automatic discovery of all event patterns and thus reducing the need for very large quantities of annotated text.

In a move towards domain portability for extraction systems, Surdeanu and Harabagiu (2002) presented an 'infrastructure for open-domain information extraction', which separates the creation of pattern matching rules into domain-independent and domain-dependent parts. Their architecture incorporates modules for recognising co-references to the same entities and events. In addition, Surdeanu and Harabagiu (2002) focus on the disambiguation of syntactic and semantic information, specifically the disambiguation of incomplete or ambiguous named-entities and nominal/pronominal references (co-

referencing of 'she', 'it', 'they').

Hasegawa *et al.* (2004) propose an entirely domain independent, unsupervised approach to IE by relaxing the requirements on the event/scenario templates to rather perform relation extraction. In this instance a relation is broadly defined as any association, such as an affiliation, role, location, part-whole, or social relation between a pair of entities. Inspired by related work on paraphrase acquisition, Hasegawa *et al.* (2004) assumed that pairs of entities that occur in similar contexts can be clustered. Assuming that each pair in a cluster is an instance of the same relation, relations that are frequently mentioned in large corpora can then be extracted.

Shinyama and Sekine (2006) expanded on the work of Hasegawa *et al.* (2004) to develop a generalised IE system based on unrestricted relation discovery. Thus a user does not need to specify which articles are relevant or the type of information required in advance. The system "pre-emptively" extracts all relations that occur frequently in the corpora and stores the relations in tabular form. The IE task is then reduced to an IR or search task, to retrieve the particular table or tables that are relevant to the user.

Sekine (2006) further expanded on the work of Shinyama and Sekine (2006), to develop a query driven IE system, using paraphrase discovery and relation extraction. Sekine (2006) combines an IR system with pattern and paraphrase discovery to create tables based on relations between entities, thus only the tables which are relevant to the user's query are presented. The prototype system developed was able to create useful tables for many topics and showed potential for domain independent IE. However, current techniques provide inadequate coverage of the extracted information (Sekine, 2006). The poor performance of language analysers, extended NE tagging, dependency analysers, and co-referencing analysers is currently holding back the expansion, portability and performance of current IE techniques (Sekine, 2006; Surdeanu and Harabagiu, 2002).

3.5.5 Information Extraction Pattern Models

Relations or events are usually extracted using an extraction pattern model. These models require a dependency analysis, in the form of a full syntactic parse of the text. For the purpose of this analysis a sentence is represented as a set of asymmetric binary links between a word and its modifiers in the form of a dependency tree (Greenwood *et al.*,

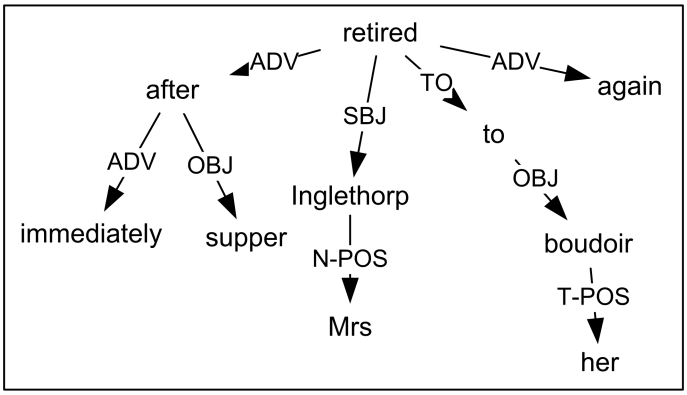


Figure 3.1: An example of a dependency tree.

2005). A word in the sentence may only modify one other word, but a word can itself be modified by many words. Figure 3.1 shows the dependency tree for the sample sentence "Immediately after supper Mrs. Inglethorp retired to her boudoir again." Appendix A describes the grammatical tags used to label the parts of the dependency tree. Meyers *et al.* (2001) describe the dependency relations between nodes and their parent, such as subject (SBJ), object (OBJ), and predicate (PRED).

An extraction pattern model is a template that defines a pattern to be extracted based on particular parts of the dependency tree. A number of extraction pattern models exist, each of which incorporates different quantities of information about the text. The extraction pattern model used determines the number of patterns capable of extraction. Sudo *et al.* (2003) introduced the use of S-expression notation to formally represent an extraction pattern model. Each node is represented in the format *a[b/c]*, or *(a:b/c)*, e.g. *SBJ[NNP/Inglethorp]*, or *(SBJ: NNP/Inglethorp)*, where c is the lexical item *(Inglethorp)*, b its grammatical tag *(NNP)* and a the dependency relation between this node and its parent *(SBJ)*. For simplicity and ease of reading, the grammatical tag b may be left out. The relationship between nodes is then represented as *(X: (A)(B)(C))*, which indicates that nodes A, B and C are direct descendents of node X.

The simplest information extraction pattern model is the predicate-argument model (Yangarber *et al.*, 2000) which is based on a direct syntactic relation between a verbal root node and its direct children or arguments. This model captures simple subject-verb-object structures, but cannot capture more complex linguistic structures such as

nominalisations or prepositional phrases. In S-expression form, the model is noted as:

(PRED: - (ARG1: -)(ARG2: -)(ARGn: -))

The following pattern can be extracted from the dependency tree in Figure 3.1 using the predicate-argument model:

- (PRED: retired (SBJ: Inglethorp))

The chains model (Sudo *et al.*, 2001) expands extraction patterns to capture indirect relationships and relationships between clausal boundaries by including elements between a modifier and its head element. The chains model defines a pattern as a path in the dependency tree passing through zero or more intermediate nodes. A limitation of this model is that it cannot represent the link between arguments of the verb. The S-expression form of this model is noted as:

(PRED: - (ARG1: - (ARG2: - ((ARGn: -)))))

The following patterns illustrate five of the nine possible patterns that can be extracted from the dependency tree in Figure 3.1 using the chains model:

- (PRED: retired (ADV: after))
- (PRED: retired (ADV: after (ADV: immediately)))
- (PRED: retired (ADV: after (OBJ: supper)))
- (PRED: retired (SBJ: Inglethorp))
- (PRED: retired (SBJ: Inglethorp (N-POS: Mrs)))

A modification of the chains model is the linked chains model (Greenwood *et al.*, 2005), which defines a pattern as a pair of chains which share the same verb, but no direct descendants. This model captures both the relationships beyond clausal boundaries as well as the link between arguments or subjects and objects of the verb. The model is defined in S-expression as:

(PRED: - (ARG1: - ((ARGn: -))) (ARG2: -((ARGm: -))))

The linked chains model generates 29 patterns from the dependency tree in Figure 3.1, including:

- (PRED: retired (SBJ: Inglethorp (N-POS: Mrs)) (ADV: after (OBJ: supper)))
- (PRED: retired (SBJ: Inglethorp (N-POS: MRs)) (TO: to (OBJ: boudoir (T-POS: her))))
- (PRED: retired (TO: to (OBJ: boudoir (T-POS: her))) (ADV: again))

The subtrees model (Sudo *et al.*, 2003) is able to capture all possible extraction patterns by considering any subtree of the dependency tree as an extraction pattern. However, the completeness of this model means that it extracts a very large number of patterns which then need to be filtered.

3.6 Visualisation for Explorative Text Mining

A large number of information visualization techniques have been developed to support the exploration of large data sets (Keim, 2002). Information visualisation alone is a massive multi-disciplinary research area encompassing many different approaches and techniques (Chen *et al.*, 2004).

Information visualisation and visual data exploration aims at integrating and involving the user in the data exploration process. Presenting the data in some visual form enables the human to use their perceptual abilities to get insight into the data, draw conclusions from the data, and interact with the data (Keim, 2002). By combining visual text mining methods with text or document structuring methods, powerful tools for the interactive exploration of document collections can be created (Hotho *et al.*, 2005).

To discuss information visualisation in any detail would be a dissertation on its own, and has been covered in many books (Card *et al.*, 1999; Chen, 2004; Ware, 2004). This section briefly mentions some of the traditional and newer graph based information visualisation techniques and provides pointers to where more information covering the work can be found. A few methods that have been applied specifically to visual text mining, are then discussed.

3.6.1 Graph Based Information Visualisation

Graphic visualisation techniques can be used to reveal the inherent relations among data elements. Graph drawing essentially consists of calculating the position of nodes within a given graph and determining the curve to be drawn for each edge (Herman *et al.*, 2000). Spatial layout and graph drawing algorithms play a fundamental role in information visualisation, as a good layout reveals the key features of a complex structure which a poor layout may obscure (Chen, 2004). However, there are a number of issues or challenges that need to be considered when choosing a graph visualisation layout or algorithm. Based on the science of perception and vision, Ware (2004) presents the key principles which result in improved clarity, utility and persuasiveness in visualisation.

The quality of the visualisation determines how much information can be conveyed and understood. Most graph drawing algorithms agree on certain aesthetic rules or criteria for what makes a drawing good (Chen *et al.*, 2004). However there is some flexibility as to which aesthetic rules should be followed for optimal human understanding (Ware, 2004; Purchase, 1997). The predictability of an algorithm has been found to be important when navigating through a graph such that two different runs of the algorithm, involving the same or similar graphs, should not lead to radically different visual representations (Herman *et al.*, 1998).

Very large graphs with hundreds of thousand of nodes impact poorly on the performance and time complexity of visualisation and a user may not be able to discern the different nodes of the graph. Thus large graphs are often reduced or clustered before traditional layout algorithms can be used (Herman *et al.*, 2000).

Battista *et al.* (1998) describe the fundamental algorithmic techniques for constructing drawings of graphs and provide an excellent introduction and coverage of layout algorithms. The tree layout is a classical and frequently used algorithm (Reingold and Tilford, 1981; Walker II, 1990) upon which a number of variants have been developed (Johnson and Shneiderman, 1991; Sindre *et al.*, 1993; Carriere and Kazman, 1995). The Sugiyama layout handles general directed graphs through a layering approach (Sugiyama *et al.*, 1981). Work on minimising the edge for this layout has been investigated (Battista *et al.*, 1998) and improvements have since been published (Martí and Laguna, 2003). Spring layouts encompass all non-deterministic layout algorithms. Often called

force-directed methods, the spring layout was first proposed by Eades (1984). This technique is based on the idea that the nodes and edges of a graph are modelled as physical bodies tied with springs. The forces between the bodies are described using Hooke's law (Herman *et al.*, 2000). This technique has been developed upon and refined by a number of researchers (Davidson and Harel, 1996; Frick *et al.*, 1995; Fruchterman and Reingold, 1991). In particular, Bertault (1999) addressed the preservation of edge-crossings to produce a more predictable algorithm. Additional layout techniques such as spanning trees (Jungnickel, 2004) and grid layouts are discussed by Battista *et al.* (1998).

A number of three dimensional algorithms have also been developed including the information cube (Rekimoto and Green, 1993), force-directed approached based on simulated annealing (Cruz and Twarog, 1995; Davidson and Harel, 1996), cone trees (Robertson *et al.*, 1991), hyperbolic layouts (Lamping and Rao, 1996; Munzer, 1997; Munzner, 1998) and a number of other techniques (Young, 1996).

3.6.2 Information Visualisation for Text Mining

Information visualisation can provide a more comprehensive and efficient means of exploring large document collections than pure text based descriptions (Hotho *et al.*, 2005) and has been motivated by the success of methods in the areas of explorative data analysis and visual data mining (Keim, 2002). According to Keim (2002), visual data exploration is especially useful when little is known about the data and the exploration goals are vague. Keim (2002) proposes that the visual exploration process can be seen as a hypothesis generation process, whereby the user may gain insight into the data by interacting with the visualisation system. By involving the user in the mining process, the exploration goals can be continually adjusted and finely tuned to come up with new hypotheses, which can then be verified either through continued exploration or using techniques from statistics and machine learning, or further controlled experimentation.

Text based data that lends itself to visualisation includes result sets, keyword relations and ontologies (Hotho *et al.*, 2005). A number of varying approaches to visualising search result sets and relations have been developed. The cat-a-cone model (Hearst and Karadi, 1997) is an interesting example of the visualisation of keyword-document relations. This model utilises a three dimensional representation of categories in hierarchies

that the user can interact with to refine a search. Sentinel (Fox *et al.*, 1999) represents documents in an abstract keyword space whereby the display is reduced down to the most important aspects of the document. The user is able to interact with and manipulate the display to view different topic aspects to identify further relevant documents. InfoCrystal (Spoerri, 1995) uses the crystal structure as a visual metaphor to visualise a Boolean query to help users deal with some of the complexities inherent in IR. Complex queries can then be created by using InfoCrystals as building blocks and organizing these blocks in a hierarchical structure (Spoerri, 1995). Havre *et al.* (2001) cluster documents based on their similarity and then visualise these clusters on lines around a circle. These clusters allow a user to visually compare multiple query result sets and explore various combinations among the sets (Havre *et al.*, 2001).

Document collections need to be visualised in a different way to result sets and keyword-document relations, due to the unstructured nature and high dimensionality of full text documents. These methods group documents based on their similarity and the visualisation process then aims to show the similarity between discovered groups of documents. Here self-organising maps, in conjunction with dimensionality reduction methods are frequently used. Flags can be used to represent a keyword or document category and colours are frequently used to indicate the density of documents in an area, or borders between different categories (Hotho *et al.*, 2005).

Lin *et al.* (1991) first proposed the use of Kohonen's self-organising map as an unsupervised learning method for the visualisation of document collections for IR. The work of Lin *et al.* (1991) was principally motivated by the work of Doyle (1961) and the idea of the similarities between Doyle's semantic map and the possible "psychological map" in the brain. This approach was then extended and improved on by Honkela (1997) and Kohonen *et al.* (2000) by incorporating zooming and colouring. Merkl (1998) proposed the use of a hierarchically organized neural network, built up from a number of independent self-organizing maps to reveal the latent document taxonomy.

Grobelnik and Mladenić (2004) presented an interesting graph based method for a collection of news articles whereby named entities that co-occur in an article are represented by relationships in a graph. Selecting a NE in the graph displays a list of keywords that frequently occur with that NE to provide additional contextual information. An

alternative approach was taken by Fortuna *et al.* (2005) for the visualisation of a document corpus. Characteristic concepts were extracted from the documents using latent semantic indexing, which were then displayed on a coloured texture document map.

A few other methods have also been applied to the visualisation of document collections. Small (1999) used an interactive clustering method to visualise scientific publications represented by a network of citations displayed in a map hierarchy. Boyack *et al.* (2002) used an approach similar to simulated annealing to construct a three dimensional landscape of a document collection. This partially interactive map is used to display a vector space description of documents in a collection or a list of directional edges, such as citation links.

3.7 Conclusion

The large amount of textual data encountered in business and personal affairs is causing an information overload for analysts who are required to read, understand, interpret, infer and estimate trends or patterns from the data. Text mining techniques enable text data analysts to explore text data, which enables the analysts to structure, find, or extract the information required more quickly and efficiently.

The fundamental limitation of text mining is that computer programs are not, and will not for a very long time, be able to read, understand and interpret text in a way that comes close to human capability (Hearst, 2003). It is also worth mentioning that, as Hearst (2003) points out, often the information to be extracted or inferred from text has not actually been recorded in textual form and needs to be deduced from other forms of communication such as voice and body language.

One of the main things holding back the usability of text mining algorithms, especially IE algorithms, is the portability of systems to different domains with minimal effort. At present, automatic IE systems require large amounts of domain knowledge to be embedded into the systems, to make these systems usable for a new domain (Sekine, 2006). Surdeanu and Harabagiu (2002) identify that domain independent IE systems using un-annotated text still face many challenges, because the performance of current language analysis tools are still poor. Thus systems are still required to be designed in

ways that require the user to interact with the system, thus requiring a large amount of time to build a new system for a new task (Surdeanu and Harabagiu, 2002).

The main culprits holding back the performance of expansion, portability and performance of current IE techniques are the language analysers which perform extended NE tagging and dependency analysis, with the main engineering bottleneck being caused by co-referencing analysers (Sekine, 2006; Surdeanu and Harabagiu, 2002). Current IE algorithms developed depend on the use of entities as anchors to recognise semantic units; therefore co-referencing (the merging of information referring to the same entity or event as well as the disambiguation of incompletely defined entities or events) plays an important role (Surdeanu and Harabagiu, 2002).

A survey of the literature shows that more sophisticated language analysis tools are needed to get better and more usable results from text mining (Surdeanu and Harabagiu, 2002; Hearst, 2003; Sekine, 2006).

When little is known about the text data or the exploration goals are vague, visual data exploration has been shown to be particularly useful (Keim, 2002). By directly involving the user in exploration, analysis goals can be adjusted on the fly. Keim (2002) suggested that the visual data exploration process can be used to assist the user to generate hypotheses, explore and gain insight into the data to validate the hypotheses and develop new hypotheses. Trends and hypotheses can then be validated through statistical or machine learning techniques.

The following chapter presents a novel framework for evidence discovery. This framework combines existing information extraction and visual data exploration techniques to create a text graph of the most important concepts and associations extracted from the text. The framework aims to address the need for more advanced tools for data analysis in digital forensics.

Chapter 4

A Framework for Evidence Discovery

Chapter 2 reviewed the current methodologies and approaches used to analyse digital data and to find and extract digital evidence. It was identified that digital forensics is not limited to cybercrimes or crimes that were committed with the use of computers or crimes committed in the virtual realm. Digital evidence can assist in an array of criminal and civil investigations ranging from fraud, identity theft, child pornography, homicides, kidnapping, abuse, and drug dealing.

The literature showed that the use of media analysis to assist with criminal investigations is currently constrained by the amount of time taken to analyse the volumes of data and the limited means of the analysis techniques available. The survey showed that there is a general call for the extension of text mining research to assist in answering the need for tools that can quickly, efficiently and accurately analyse data to reveal truly useful information.

One of the primary difficulties with finding relevant information quickly using the current tools and methods is that they tend to be query- or search-term driven and these terms, by definition, rely on the analyst having prior or background knowledge about the data to be analysed. Accordingly, search term analysis is necessarily restricted to the portions of data relating to the query terms. No information is obtained about data which does not fit the query terms. To make the investigation process more efficient,

tools need to be developed that will reduce the quantity of data to be analysed, aid the analysts' exploration of the data and enhance the intelligibility of the presentation of the data.

Chapter 3 surveyed the significant literature in text mining. Text mining techniques enable text data analysts to explore text data, which enables them to structure, find, or extract the information they require more quickly and efficiently.

At present, automatic information extraction systems require large amounts of domain knowledge to be embedded into the system to make the system usable for a new domain (Sekine, 2006). The large quantity of domain knowledge required makes the portability of text mining systems to different domains extremely limited thus holding back the usefulness of text mining and, in particular, information extraction systems. Surdeanu and Harabagiu (2002) identified that domain independent information extraction systems using un-annotated text still face many challenges. This is because the performance of current language analysis tools remains poor, causing system designs to require significant user interaction and making the design of a new system a time-intensive task (Surdeanu and Harabagiu, 2002). The literature showed that more sophisticated language analysis tools are needed to get better and more usable results from text mining.

Visual data exploration was identified to be particularly useful when little is known about the text data or when the exploration goals are vague. The visual data exploration process can be useful in assisting the user to generate, develop, and validate hypotheses, and to explore and gain insight into the data. Trends and hypotheses can then be validated through statistical or machine learning techniques.

This research aims to address the need for more efficient data analysis tools for digital forensics through the adaptation and application of current text mining methods. This dissertation hypothesizes that information extraction techniques combined with visual exploration techniques can assist in identifying suspects and events, and the relations between these entities which could assist an investigator to piece together the story surrounding the crime, create hypothesises for potential leads for further investigation, or identify pieces of data which could form useful supporting evidence in a trial.

This chapter proposes a novel framework in which to perform evidence discovery

in an attempt to meet these challenges. The framework combines existing information extraction and visual data exploration techniques to create a text graph of the most important concepts and associations extracted from the text. The text graphs provide an overview and general representation of the text, which may help the investigator explore and analyse the text more efficiently. An overview of the framework is first described, followed by descriptions of each component of the framework namely, document pre-processing, relation discovery, and text-graph creation and visualisation.

4.1 Overview of Evidence Discovery Framework

It is not uncommon in digital investigations for very little to be known about the case or the collected data prior to analysis. A discovery system which does not rely on user input for query terms could therefore assist in speeding up the analysis phase of the investigation.

Due to the sensitive nature of case datasets, it is difficult, especially in the South African context, to obtain authentic data for research purposes. De Waal *et al.* (2008) performed a case study on an authentic forensic dataset, however the usefulness of their technique could not be fully evaluated as the investigators who provided the dataset had not themselves generated usable evidence from the data. For these reasons a fictitious dataset was used in this research to establish the feasibility of the framework and subsequent evidence discovery system.

The murder mystery novel by Agatha Christie entitled *The Mysterious Affair at Styles* was chosen as a dataset as this novel is available for free download in electronic plain text format from Project Gutenberg (Lebert, 2008). An Agatha Christie novel was chosen because it was felt that it represents the complex wealth of knowledge that is present in real evidentiary data. Agatha Christie's novels are characterised by complex plots involving a crime and several characters, all or many of whom are typically presented as potential suspects in the reader's mind. Christie's mysteries are achieved by reserving the identity of the true criminal or criminals until the conclusion of the novel. Because the stories are complex and so many characters are potential suspects, there is an element of noise surrounding the main thread of the crime. Although a literary work

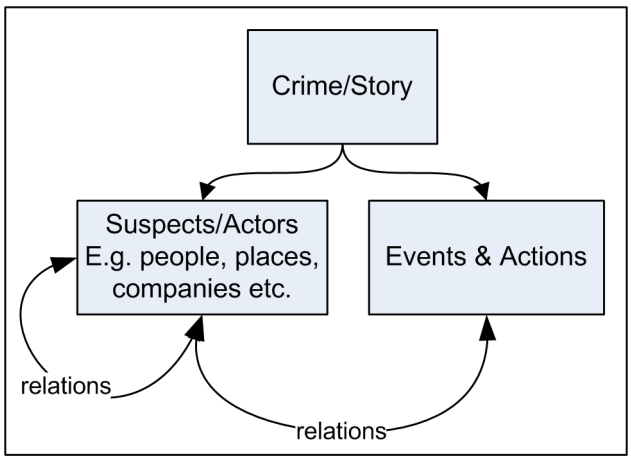


Figure 4.1: Parallels between a crime and a story.

is not written in the same style as business related documents, letters and memos, the datasets used to develop linguistic analysers, parsers and information extraction algorithms, generally comprise a very large collection of short news articles. Using a literary work will therefore test the portability of the existing NLP tools, which will give an indication of their potential performance on a real criminal dataset.

Like a novel, a true-life criminal act and the events and suspects surrounding it form a story, or narrative, consisting of actors, actions and relations. Figure 4.1 shows the parallels between a crime and a story.

Visualising extracted concepts and/or NEs, and their relations in a text-graph, could assist an investigator to piece together the story surrounding the crime. Thus, relation extraction techniques seem apt for the challenge. By removing the reliance on an input query as required by traditional IE systems, the evidence discovery system enables the investigator to explore portions of the data that the investigator would not be able to using traditional search based methods.

The evidence discovery system proposed in this dissertation consists of a few pre-processing steps followed by relation extraction, which is then used to create text-graphs of the story. Figure 4.2 shows the components of the framework and the flow of data.

The following sections describe each component of the framework illustrated in Figure 4.2. First the text documents need to be processed with some basic text analysis. Relation extraction techniques are then employed to extract concepts from the text and

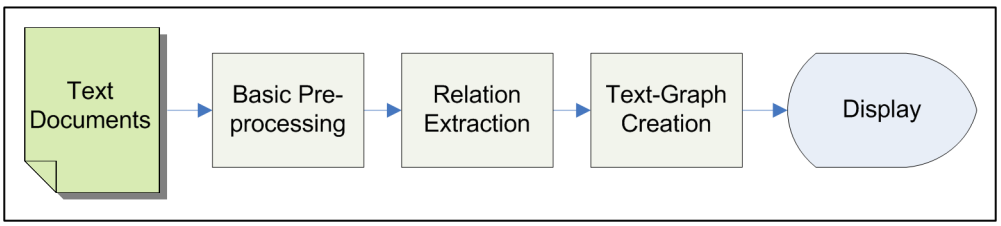


Figure 4.2: Overview of the Evidence Discovery Framework.

their relations or 'associations' to each other. A text graph is then created from the extracted concepts and relations, which is then visualised and presented to the user in an interactive interface, which enables the user to explore and evaluate the associations in the graph.

4.2 Document Pre-processing

Text pre-processing algorithms are used to convert text from a simple sequence of character strings into tokens and chunks so as to identify syntactic elements and to store the data in a form that is more appropriate for further processing than a plain text file. Different types of pre-processing can be done, and the appropriate pre-processing techniques should be chosen depending on the text analysis algorithm which is intended to be used.

The relation extraction algorithms used in this thesis require two forms of text pre-processing. First the text documents are processed by a part of speech tagger, using Infogistics' NLProcessor software (Infogistics, 2001). NLProcessor first tokenises the text, segmenting the stream of characters into separate parts or tokens to obtain the basic units of paragraphs, sentences and words. An algorithm is then applied to determine the part-of-speech of each word in a sentence based on its definition and context and applies a label or tag (e.g. noun, verb, adjective, etc.) for each word, based on the Modified Penn Treebank Tag-Set (see Appendix A). Figure 4.3 shows an example of a sentence that has been tagged with part of speech tags. WordNet (Fellbaum, 1998) is then used to tag each word with its lemma, based on its part of speech to represent each word in its normalised form.

```
#SRC:  Immediately after supper, Mrs. Inglethorp retired to her boudoir again. ``Send my
coffee in here , Mary," she called.

Immediately_RB after_IN ([ supper_NN Mrs._NNP Inglethorp_NNP ])
<: retired_VBD :>
to_TO ([ her_PRP$ boudoir_NN ]) again_RB ._.

"_``
<: Send_VB :>
([ my_PRP$ coffee_NN ]) in_IN here_RB ,_, ([ Mary_NNP ]),_, "_`` ([ she_PRP ])
<: called_VBD :>
._.
```

Figure 4.3: Example of sentence pre-processing.

4.3 Relation Discovery

After the text has been pre-processed and marked up with syntactic tags, these tags can be used to perform further text analysis. Relation discovery aims to find and extract concepts and their relations or associations.

This thesis will present two relation discovery models. The first (Model A) relies on statistical measures of association and co-occurrence, whereas the second (Model B) aims to exploit the syntactic structure and linguistic characteristics of the text, such as NER and full syntactic parsing. These two relation discovery models are discussed in detail in the following chapter.

Normally, extracted information would be stored in tabular form, where events or similar information would be grouped together in tables. The relevant table can then be retrieved when desired, using a query. For example, a query with the key words “merge, merger, acquisition, purchase, buy” may produce a table in the format shown in Table 4.1.

In this research the extracted concepts and relations are stored in the form of text graphs to explore the data, rather than to retrieve it. The text-graphs can then be visualised and explored.

Table 4.1: Example of a typical table retrieved from an IE query.

Company	Money	Date
ABC Bank	about $1.6 billion	
BB Corp, XYZ inc.	$3 million	last month
JJ Holdings	$286 million	
World Finance Corp.	about $400 million	17 July

4.4 Text Graph Creation and Visualisation

Visualisation is a very important aspect of any system that is intended to be used by a human, as humans are very receptive to interpreting graphical information. To provide a means for the user to explore and evaluate the extracted relations, text graphs are created from the concepts and their relations. These text graphs are then visualised and presented to the user in an interactive interface.

The text graphs are created in the form of a highly connected graph where nodes represent concepts and links are drawn to show associations between concepts. The graph is then displayed using a force-based algorithm (see Appendix C) on a graphical user interface (GUI). Figure 4.4 illustrates resultant text graphs created from the entire novel used for this example.

'Key Concepts' are defined to be those concepts that are important to the story based on the fact that they are highly connected. Nodes that have more than a specified threshold, n_k, links are highlighted as 'key concepts' to enable the user to see at a glance which concepts are important to the story in the novel. The number of relations or links that a node has can be used as an indicator of that node's importance within the text. The higher the number of links, the more important the node is. A histogram of the number of links for the nodes extracted from the novel in this example, for each of the relation extraction methods, is presented in Graph 4.1. The histogram shows that the greater the number of links, the fewer the number of nodes having that many links. There is a rapid fall-off of the number of nodes as the number of links increases, which aids in choosing the threshold, n_k. For the purposes of this case study a threshold of $n_k \geq 4$ was chosen through trial and error, and can be optimized for corpora of varying

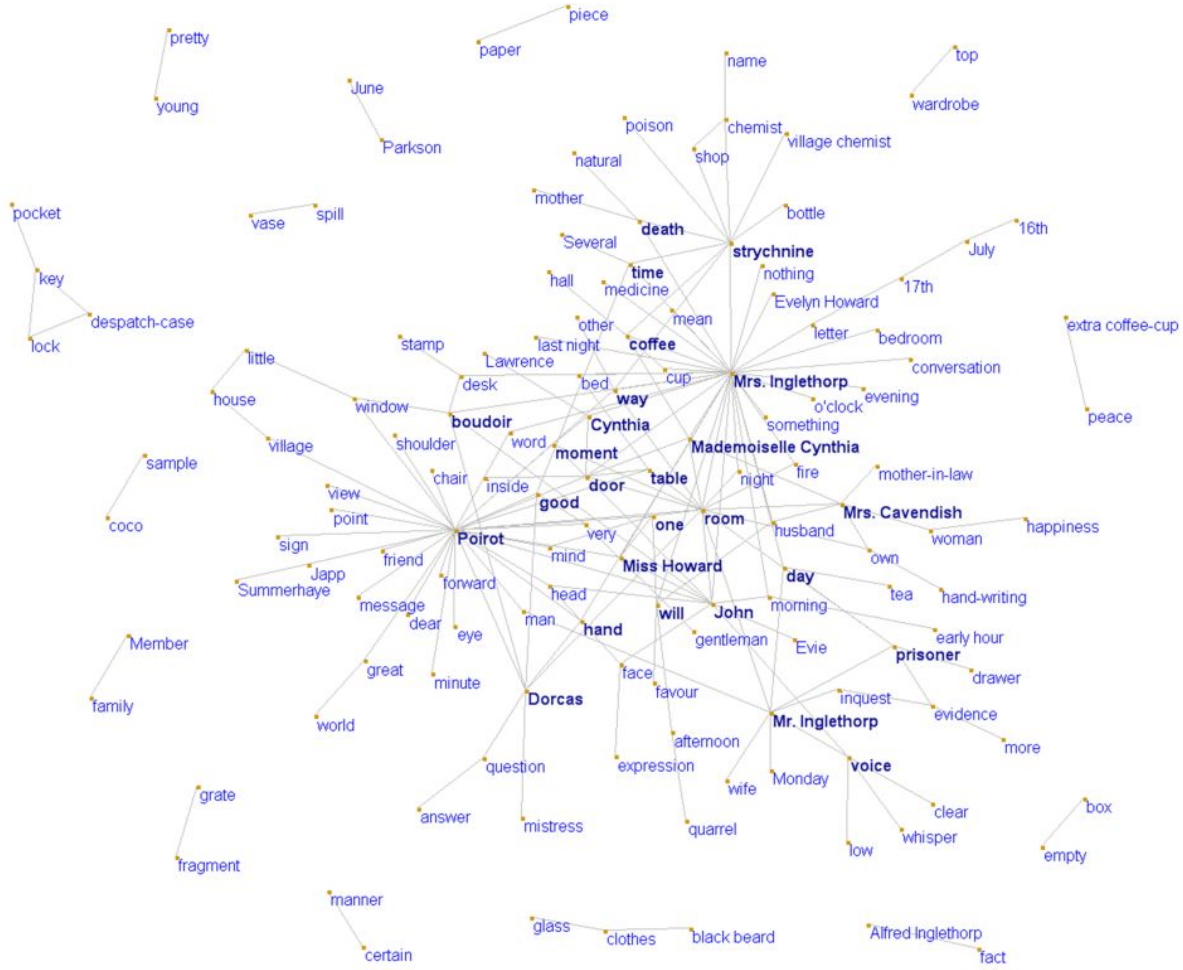
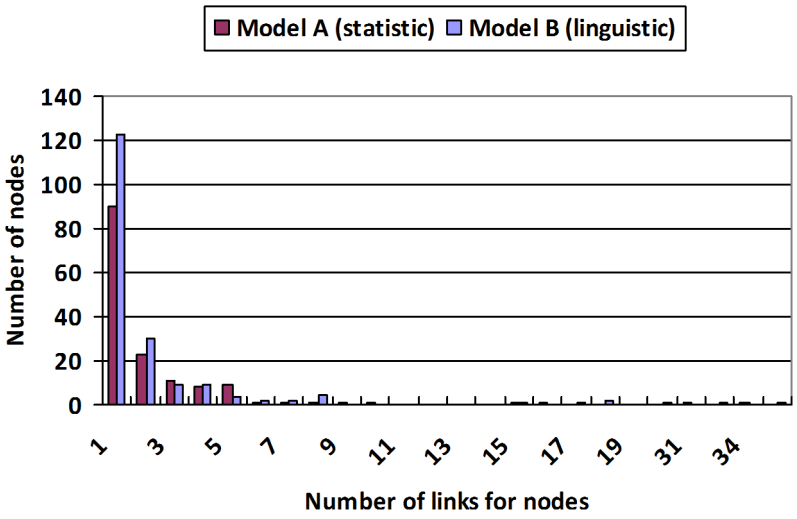


Figure 4.4: Visualisation of the Text Graphs created from the novel *The Mysterious Affair at Styles.*

sizes depending on the users' preference.

To understand the graph in Figure 4.4 better, the user is able to interact with it in two ways. Firstly, where there are an immense number of nodes, the user may rearrange the nodes to organize, cluster, separate or spread out certain nodes in the graph. This is done by clicking on a particular node and dragging it to the desired position. Secondly, the user may explore the context of the links. When a single node is selected a side panel on the GUI shows all of the sentences that created the association between that node and all the concepts that the node is connected to. If two nodes are selected, only



Graph 4.1: Histogram of number of links for the nodes extracted from the example.

the contexts of those links between those two nodes are shown. This helps the user to understand the meanings of the concepts as well as their association with each other. Figure 4.5 demonstrates this functionality. By traversing the links, an investigator may explore relations between concepts that are connected through other concepts. The degree of separation is measured by the number of links between the two concepts. Thus threads of the story can be explored.

Model B of the relation discovery process identifies entities and NEs. If entities have been identified and labeled in the graph, the GUI will display a list of these entities in the side panel. The user may then select an entity from the list to display a subgraph containing the selected entity and its links. In this way a user may examine entities and their links more closely. Figure 4.6 shows the selection of the *person* entity "Poirot (PER)" from the list and the corresponding subgraph.

4.5 Conclusion

The framework for evidence discovery proposed in this chapter combines information extraction techniques with visual exploration techniques to provide a novel means to perform evidence discovery. By utilising unrestricted, unsupervised information extrac-

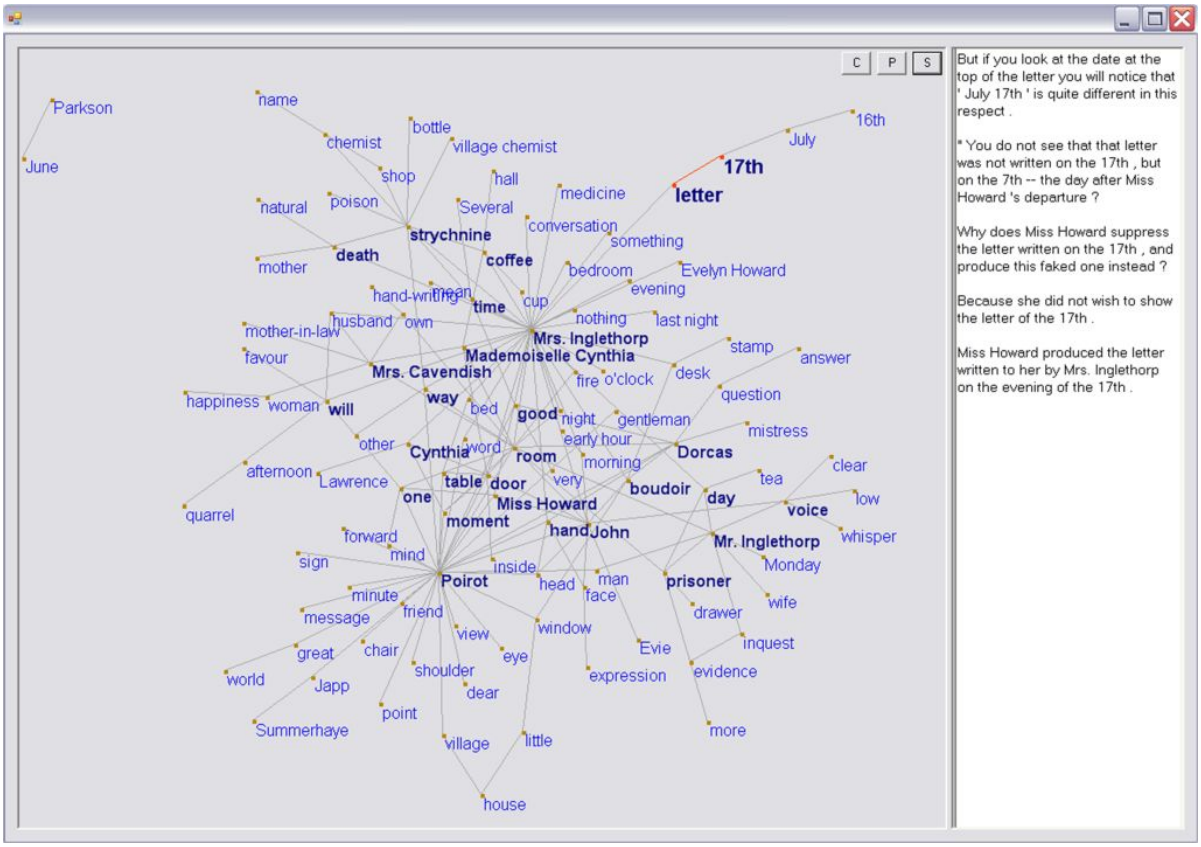


Figure 4.5: Side panel shows context of links between "letter" and "17th".

tion techniques, the investigator does not require input queries or keywords for searching, thus enabling the investigator to analyse portions of the data that may not have been identified by keyword searches.

Text graphs of the most important concepts and associations extracted from the full text establish ties between the concepts and provide an overview and general representation of the text. Through an interactive visual interface the investigator can explore the data to identify suspects, events, and the relations between suspects. A text-graph of this nature enables the investigator to discover salient relationships by traversing the links to discover 2^{nd} to n degree relations. The text graphs assist the investigator to piece together the story surrounding the crime, create hypotheses for potential leads for further investigation, or identify pieces of data which could form useful supporting evidence in a trial, without having to read the entire text.

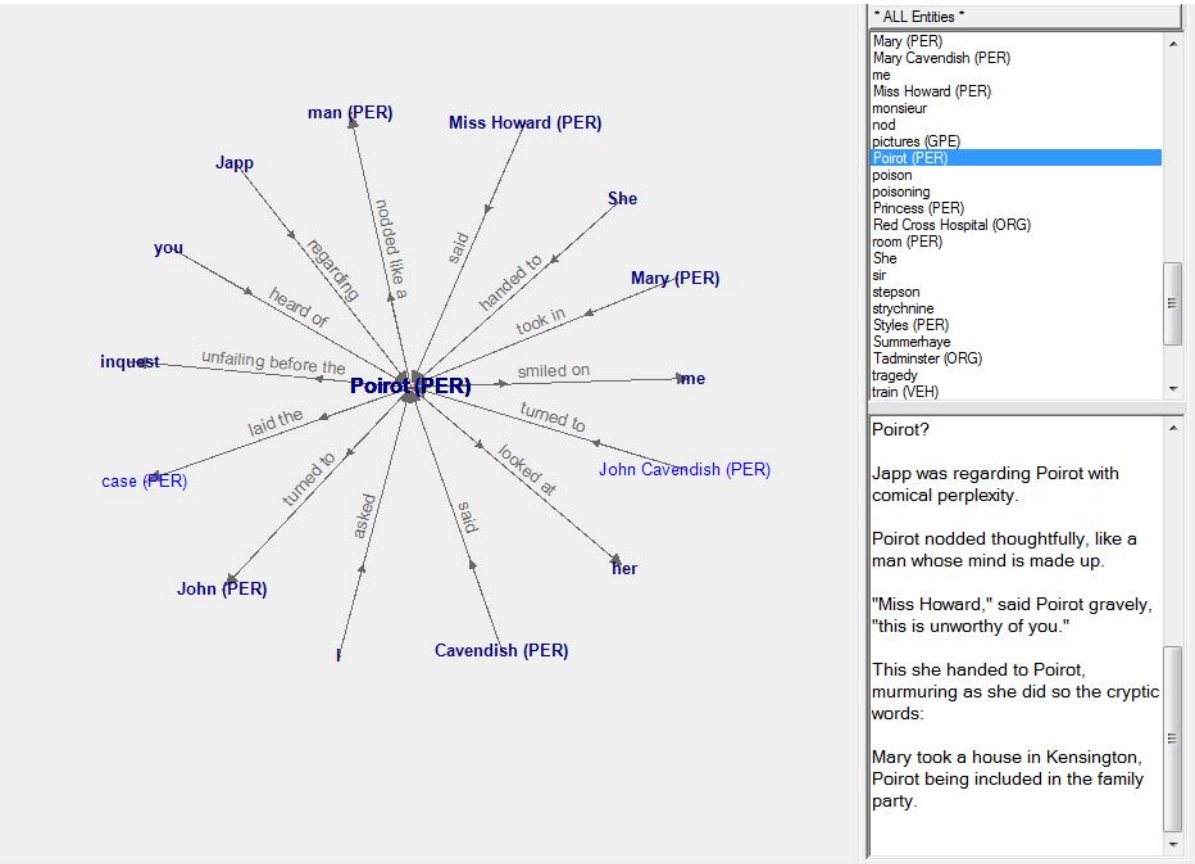


Figure 4.6: Selection of entity "Poirot" and its corresponding subgraph.

The following chapter presents two models for performing the relation extraction process of the framework. The results of these models applied within the framework are then discussed and compared in Chapter 6.

Chapter 5

Relation Discovery Models

The previous chapter presented a novel approach to performing text analysis for evidence discovery. That chapter presented a framework for evidence discovery which consisted of four processes. Different approaches can be taken to find and extract relations between concepts from textual data. This chapter presents two models for performing the relation extraction process of the framework.

The first model takes a statistical approach to discovering relations based on co-occurrences of complex concepts (Model A). The details of this approach are discussed in Section 5.1. Model B utilises a linguistic approach using NE extraction and IE patterns. The details of this model are described in Section 5.2.

The results of these models applied within the framework presented in Chapter 4 are discussed and compared in Chapter 6.

5.1 Model A: Relation Discovery and Extraction through Co-occurrence of Concepts

A concept is an "idea" or "unit of thought" and is said to be complex if it consists of more than one word. A complex concept is formed from co-occurring words that are strongly associated with each other.

This model takes a statistical approach to discovering relations based on co-occurrences of concepts. Model A utilises a 'bag of words' approach based on the assumption that a

relation occurs between two concepts if these concepts co-occur within a certain boundary or window with a frequency above a certain threshold. The following sub-sections describe the creation of complex concepts and the relation extraction process.

5.1.1 Creation of Complex Concepts

Complex concepts are generated based on the term generation approach of Feldman *et al.* (1998) and Daille (1996). Feldman *et al.* and Daille define a term as a series of one or more lemmas (words in their base form) selected based on their morpho-syntactic pattern (part-of-speech patterns). Model A uses the following patterns: "noun noun", "noun preposition noun", and "adjective noun." All adjacent words that fit one of these patterns are extracted as candidate terms. These candidate terms are then filtered based on a statistical score and the remaining candidate terms are combined and replaced in the text as one unit or word.

The combination process is performed in several passes, as this enables the system to calculate the statistical score for each pair in each pass. Competing possibilities such as (box/noun of/prep chocolate/noun) and (chocolate/noun treats/noun) in (box/noun of/prep chocolate/noun treats/noun) are resolved by combining the pair with a better statistical score first. This process is repeated until no new pairs are combined. Algorithm 5.1 describes the method/process implemented to form these complex concepts.

The statistical scoring employed in most term extraction methods use a combination of co-occurrence frequency and at least one other statistical measure most suited for the applicable corpus. Co-occurrence frequency is the simplest measure and is determined by the number of times the two words occur together. For the purposes of model A, co-occurrence frequency was used in conjunction with the association ratio, see equation (3.6). Since the association ratio shows very little significance for pairs with a very low frequency, a threshold $T_{freq} = 4$ was set to filter out pairs that are not significant. A very low threshold was chosen in this case, because the corpus is relatively small. Raising the threshold will increase the association strength required for the pair to be considered as significant, which may be required depending on the size and nature of the corpus.

Since only adjacent terms that fit the morpho-syntactic criteria stated above were of interest, a default window of size 2 was chosen. When $MI(x, y) \approx 0$, the probability

```
double dAssocThresh; //threshold for the association ratio
int iMinNumOcc; //threshold for the number of occurances

pairComplexConcepts( dAssocThresh, iMinNumOcc )
{
    findCandidateConcepts(); //find all words matching the POS patterns
    while (m_candidateConcepts.Count > 0) //perform algorithm over several passes
    {
        calcConceptAssocValue(); //calculate the association ratio for each concept
        filterConcepts( dAssocThresh, iMinNumOcc );//filter concepts based on thresholds
        replaceWordsWithConcepts(); //insert concepts back into text
        findCandidateConcepts(); //find pairs missed in previous pass
    }
}
```

Algorithm 5.1: Creation of Complex Concepts

that the two words occur together by chance is almost equal to the probability that their co-occurrence is due to a genuine association. As $MI(x, y)$ increases, the confidence that the pair of words should be combined into a single term increases. A threshold, $T_{assoc} = 3$ is set to filter out pairs that are not regarded as being associated. This threshold was chosen through trial and error and varies according to the size of the corpus. A very low threshold will increase the number of terms combined, where a high threshold will only combine terms that occur together at high frequencies. Table 5.1 shows a selection of the results from the term extraction process applied to the entire novel. Results are ranked first by the association ratio and then by co-occurence frequency.

Examination of the resultant complex conepts show that these are indeed sensible. Descriptive words in the complex concepts give additional information such as 'Aunt' in 'Aunt Emily', 'at Tadminster' in 'Hospital at Tadminster' and 'Dr.' in 'Dr. Wilkins'. In the same way these complex concepts give more interest to terms that are combined such as 'bunch of keys', 'black beard', 'violent quarrel', and 'last link'.

Table 5.1: Example of Complex Concepts.

Combined Terms	Num. Occ.	Freq.	Assoc. Ratio
Red Cross	4	0.01	13.557404
Scotland Yard men	6	0.01	11.923638
bunch of keys	4	0.01	11.634572
Aunt Emily	5	0.01	11.618804
Wilful Murder	4	0.01	11.618804
black beard	11	0.02	11.372979
Sir Ernest Heavywether	6	0.01	11.287145
intense suprise	4	0.01	10.972441
Mr. Lawrence Cavendish	7	0.01	10.933037
Hospital at Tadminster	4	0.01	10.732975
French window	4	0.01	10.34 7951
important papers	4	0.01	10.247549
violent quarrel	4	0.01	10.097972
few minutes	11	0.02	10.000027
Dr. Wilkins	15	0.02	9.8471331
last link	7	0.01	9.7797963
Dr. Bauerstein	47	0.06	9.7203911

5.1.2 Relation Extraction

As stated above, model A states that a relation occurs between two concepts if these concepts co-occur within a certain boundary or window with a frequency greater than a defined threshold. Based on the idea of market basket analysis whereby associations are found between purchased products, model A aims to find 'association links' or relations between the frequently occurring concepts in the text.

A relation is said to exist between two concepts if the concepts co-occur within a certain window size or text segment having a support of a chosen minimum number of occurrences. To form relations that make semantic sense, concepts were filtered so as to only include concepts tagged as nouns and adjectives.

The size of the text segment used can be varied in order to optimise the results of

Table 5.2: Sample of the Extracted Relations (Model A).

Concept 1	associated with	Concept 2
afternoon	$\Longleftrightarrow$	will
coffee	$\Longleftrightarrow$	Strychnine
Cynthia	$\Longleftrightarrow$	Lawrence
Cynthia	$\Longleftrightarrow$	moment
death	$\Longleftrightarrow$	Strychnine
death	$\Longleftrightarrow$	mother
favour	$\Longleftrightarrow$	will
Mrs. Inglethorp	$\Longleftrightarrow$	Strychnine
one	$\Longleftrightarrow$	will

the text graphs. For the novel used in this work, segments consisting of one sentence were implemented, as this causes a strong association between the concepts. Increasing the size of the text segment will increase the distance between concepts and lessen the degree of association enforced. A sample of the relations extracted is shown in Table 5.2.

Examination of the associations shows that these are indeed sensible. "Afternoon" and "will" occur together five times and in each case the question of a will being drafted on the afternoon in question is discussed. The association between "coffee" and "strychnine" is intriguing and is summed up by one of the sentences in the novel, which states: "The present contention is that Mrs. Inglethorp died of strychnine poisoning, presumably administered in her coffee." Exploration of "Cynthia" and "Lawrence" merely shows that they are acquaintances, however in some cases a discovery of that nature could be the key to the mystery or crime.

5.2 Model B: Relation Discovery and Extraction using NEs and IE patterns

Model B utilises a linguistic approach and aims to extract relations using NE extraction and IE patterns. The components of this model and flow of data are shown in Figure 5.1.

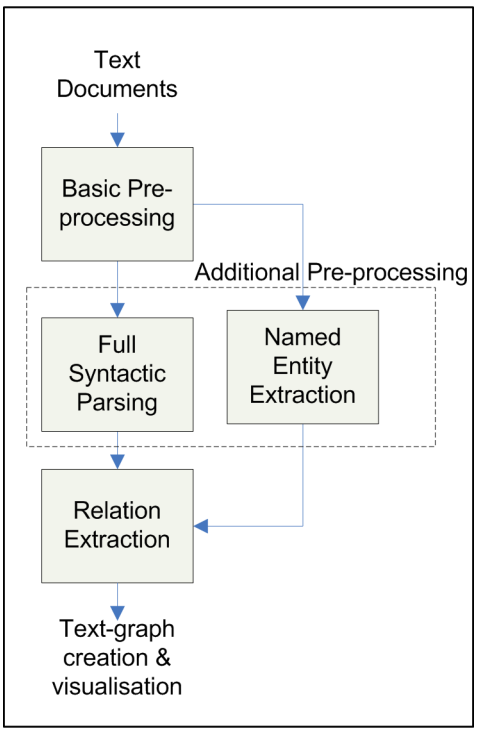


Figure 5.1: Components of the Relation Discovery using NEs and IE patterns.

To make use of the syntactic structure to extract relations some additional linguistic pre-processing steps, namely full syntactic parsing and NE extraction, are required. After the basic pre-processing, whereby the text is split into sentences and words, shallow parsing is performed using the Charniak parser (Charniak, 2005) which performs part of speech (POS) tagging and phrase chunking. The parsed text is then subjected to NE extraction and full syntactic parsing. In this dissertation, NE extraction was performed using the Java Extraction Toolkit (JET) (Grishman, 2007). The extracted NEs are put aside for later use in the relation extraction process.

In order to extract relations using an extraction pattern model, a dependency analysis, in the form of a full syntactic parse, needs to be performed. For the purpose of this analysis a sentence is represented as a set of asymmetric binary links between a word and its modifiers in the form of a dependency tree annotated with its grammatical type and its function in the sentence, e.g. subject, object, etc. This study uses a syntactic parser developed at New York University, the Grammatical and Logical Argument Framework

(GLARF) (Meyers *et al.*, 2001), to perform full syntactic parsing. GLARF is a typed feature structure framework for representing regularizations of parse trees. GLARF produces dependency trees with both the surface and logical relations. Where sentences are written in the active voice the surface and logical relations are the same, however in the passive voice the true relations are represented in the logical form. For example, in the sentence "The apple was eaten by John," the surface subject is "the apple" and the logical subject is "John." Therefore the logical relations are used for relation discovery. Figure 3.1 shows the dependency tree for the sample sentence "Immediately after supper Mrs. Inglethorp retired to her boudoir again."

The extracted NEs in conjunction with the dependency trees are then used to extract relations using an IE pattern model. The relation extraction process is discussed in the following sub-section.

5.2.1 Relation Extraction

Information extraction pattern models were introduced and discussed in Chapter 3. An extraction pattern model was defined as a template that defines a pattern to be extracted based on particular parts of the dependency tree. A number of extraction pattern models exist, each of which incorporates different amounts of information about the text. The extraction pattern model used determines the number of patterns that can be extracted.

The most commonly used pattern models and their pros and cons were described in Chapter 3. The linked chains extraction pattern model was chosen for the evidence discovery system because it offered the best compromise between capturing too many patterns or too little information. The linked chains model (Greenwood *et al.*, 2005) defines a pattern as a pair of chains which share the same verb, but no direct descendants. The linked chains model captures both the relationships beyond clausal boundaries as well as the link between arguments or subjects and objects of the verb[1].

From the dependency tree in Figure 3.1, a total of 29 linked chains can be extracted. For the purposes of relation extraction for the evidence discovery system, relations of

[1]The S-expression form of the linked chains model is noted as (PRED: - (ARG1: - ((ARGn: -))) (ARG2: -((ARGm: -)))).

the form "subject-relation-object" were chosen to ensure that relations contain concepts rather than other parts of speech such as adjectives or adverbs. Thus only patterns that contain both a subject and an object are extracted. This results in only two linked chains from the dependency tree in Figure 3.1:

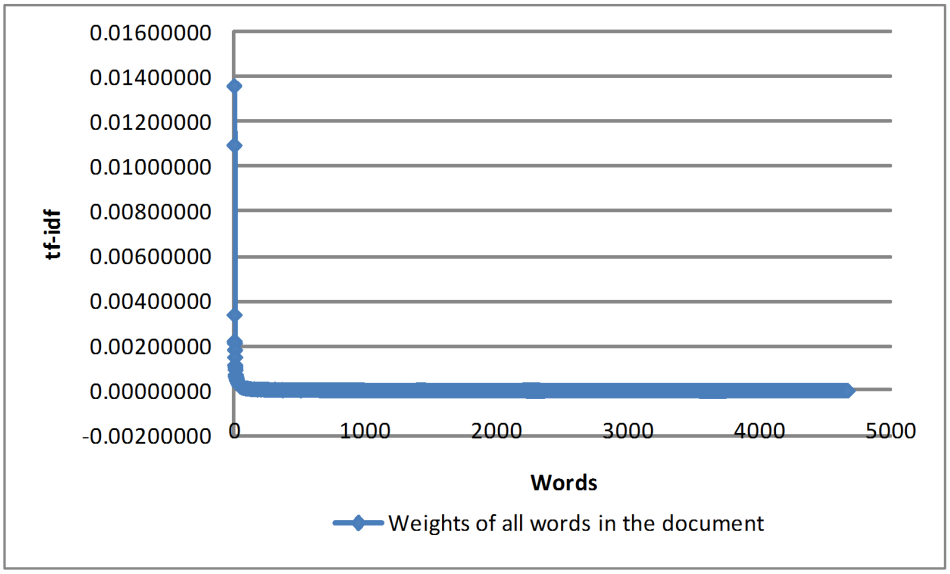
- (PRED: retired (SBJ: Inglethorp (N-POS: Mrs)) (ADV: after (OBJ: supper)))
- (PRED: retired (SBJ: Inglethorp (N-POS: MRs)) (TO: to (OBJ: boudoir (T-POS: her))))

However, it was found that the syntactic parser could not always parse the entire sentence and would leave fragments, some of which did not contain a verb, and therefore no patterns could be extracted from those fragments.

The extracted NEs were then inserted back into the extracted relations as tags to indicate their identity and class. An explanation of the NE tags used in this dissertation is presented in Appendix B. The co-referencing of NEs is a difficult task for a novel or literary work with many characters, especially where names are shared and are often refered to using pronouns. It was found that the number of errors in the co-referencing of NEs were more harmful than helpful. Therefore, it was decided that the NEs would be tagged, but not co-referenced.

At least one relation can be extracted for each sentence in the text, which results in too many relations for the user to be able to grasp. Thus weights were applied to each subject and object of the extracted relations using the tf.idf weight described in Section 3.2. The tf.idf is a statistical measure used to evaluate how important a word or term is to a document in a collection or corpus. Only relations whose subject and object are ranked as important were retained, reducing the number of relations to a small set of important relations, which is more manageable for the user. The tf.idf was calculted using Equation (3.3). The document collection was assembled from 80 books downloaded from the Project Gutenberg (Lebert, 2008) repository.

Graph 5.1 shows the weights of all of the words in the text in descending order. The higher the weight the more important the word is to the document. In order to determine wich words are siginificant and should be retained, a threshold value for the tf.idf was chosen. The threshold was chosen to eliminate 98% of the values by calculating the mean

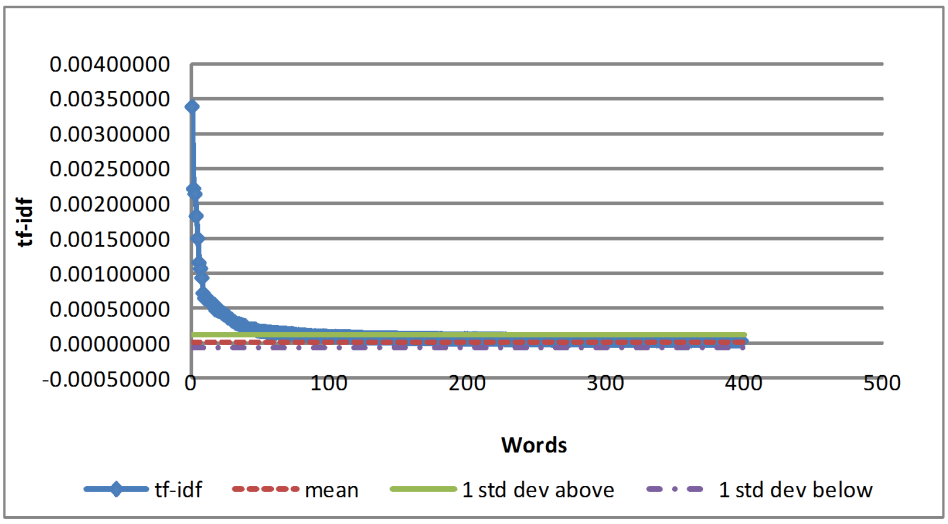


Graph 5.1: Tf.idf weights for all words in the document.

and standard deviation of the weights and setting the threshold to one standard deviation above the mean. Graph 5.2 shows these values. The two highest weights, assigned to the two main character's names, were found to be extreme values because they occured very frequently in the relevant book and did not occur in any of the other books in the collection. For that reason those two weights were excluded from the calculation of the threshold. Thus the threshold was set to a value of $1.0856E^{-4}$. Terms that were identified as NEs were assigned a weight of 1.0 to ensure that they are not filtered out. For example, the name "John" may have a very high document frequency causing a very low tf.idf. However, NEs are assumed to have a high evidentiary value to investigators and therefore they are assigned a very high weight. Only relations whose subject and objects were weighted above the threshold were then retained.

Tables 5.3 and 5.4 show a sample of the extracted relations from the novel used in this dissertation. Table 5.3 shows the relations before filtering, whereas Table 5.4 shows the reduced number of extracted relations after the NEs have been tagged and inserted back into the text.

The resultant relations, after filtering, are shown to be sensible relations between the subjects and objects, despite the removal of words from the original sentence in the extraction process. Although some errors occured in the NE tagging, the relations in-



Graph 5.2: Tf.idf threshold calculated for the document.

volving NEs give the reader information about the NEs and can describe the associations between NEs. Relations that do not include any NEs are shown to be interesting and add detail to the story behind the data.

Text-graphs were then created to illustrate the ties between the extracted relations based on common NEs, subjects, and objects. The subjects and objects of the relations are represented by nodes, with labelled directed links representing the action of the relation using an arrow from the subject to the object. Multiple relations between nodes may lead to more than one link between two nodes. Figure 5.2 shows an example of a text-graph created from a selection of extracted relations from the text. This example shows that the nodes 'Inglethorp' and 'John' are both of type person (PER) and are both related to the object 'room'. Their relations with 'room' are 'had in the' and 'opened door of' respectively.

5.3 Conclusion

This chapter presented two models for performing the relation extraction process of the evidence discovery framework. By working within a framework, individual components can be worked on and improved in a comparable manner. The results of these models applied within the framework are discussed and compared in the following chapter.

Table 5.3: Sample of the extracted relations with NE tagging and tf.idf weights (Model B).

Subject	tf.idf	Relation	Object	tf.idf
Interest	$6.9885E^{-6}$	aroused in the	public (PER)	1.0
I	$1.68E^{-4}$	write account of	story	$1.0944E^{-5}$
Cavendish (PER)	1.0	in mother-in-law's	room (PER)	1.0
Cavendish (PER)	1.0	purchased Their	country-place	0.0
I	$1.68E^{-4}$	Tell	you	$4.5510E^{-4}$
coco	$6.2134E^{-4}$	contained no	strychnine	$1.4912E^{-3}$
John (PER)	1.0	views concerning	Bauerstein	$2.1277E^{-3}$
Saucepan	$6.0113E^{-5}$	is in Inglethorp's	room (PER)	1.0
She	$1.9426E^{-4}$	said to	John (PER)	1.0
Japp	$7.1129E^{-4}$	accompanied	car (VEH)	1.0
I	$1.68E^{-4}$	descended from the	train (VEH)	1.0
step-mother	$2.1212E^{-5}$	generous to	them	0.0
fellow	$2.3939E^{-5}$	useful to	her	$1.3390E^{-4}$
Cynthia (PER)	1.0	protegee of the mother	daughter (PER)	1.0
man (PER)	1.0	followed in his	manner	$9.9622E^{-6}$

Table 5.4: Extracted relations from above after filtering (Model B).

Subject	tf.idf	Relation	Object	tf.idf
Cavendish (PER)	1.0	in mother-in-law's	room (PER)	1.0
I	$1.68E^{-4}$	Tell	you	$4.5510E^{-4}$
coco	$6.2134E^{-4}$	contained no	strychnine	$1.4912E^{-3}$
John (PER)	1.0	views concerning	Bauerstein	$2.1277E^{-3}$
She	$1.9426E^{-4}$	said to	John (PER)	1.0
Japp	$7.1129E^{-4}$	accompanied	car (VEH)	1.0
I	$1.68E^{-4}$	descended from the	train (VEH)	1.0
Cynthia (PER)	1.0	protegee of the mother	daughter (PER)	1.0

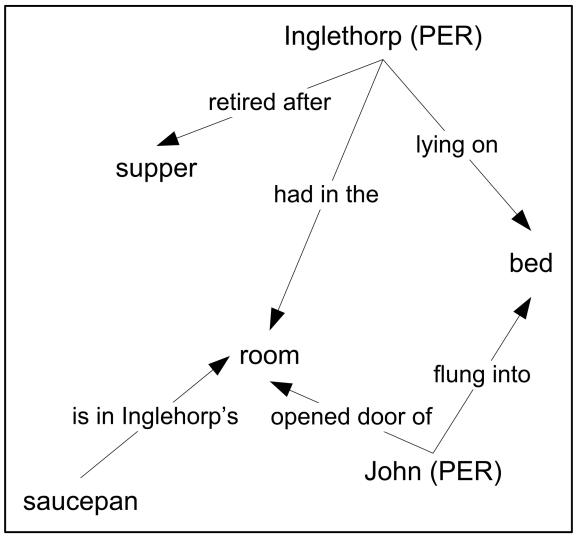


Figure 5.2: Example of a text-graph created from extracted relations.

Chapter 6

Experimental Design and Results

A novel approach to performing text analysis for evidence discovery was presented in Chapters 4 and 5. Chapter 4 presented a framework for evidence discovery which consisted of four processes. Different approaches can be taken to find and extract relations between concepts from textual data. Chapter 5 presented two models for performing the relation extraction process of the framework. Model A utilised a statistical approach to discovering relations based on co-occurrences of complex concepts. Model B utilised a linguistic approach using NE extraction and IE patterns. The evidence discovery system gives a graphical output in the form of graphs, which represent concepts and their relations.

Some method is needed to assess whether the product of the framework and models is useful and whether the graphical presentation of the data is useful to a user. If neither of the models produce results that are useful to a user, then it must be concluded either (a) that the novel approach to text analysis for evidence discovery presented here is not a successful/viable one or (b) that the implementation of that approach was not successful. If either of the models do produce results that are useful, then it can be argued that the approach advocated here can be successfully applied to the analysis of textual data for digital forensics.

The purpose of this chapter is therefore to evaluate the useability or usefulness of the results produced by each of the models. Because a full-scale quantitative assessment of the performance of the models is beyond the scope of this study, a preliminary study

was devised with the aim of determining whether the evidence discovery system produces sensible, useful and useable results.

A survey of the literature on digital forensics presented in Chapter 2, showed that digital forensic analysis is predominantly performed using text string searching (Beebe, 2007). Text based searching therefore provides a useful comparator or control against which to compare the two models advocated here. This chapter discusses the design of such an experiment, followed by the methodology that was carried out. The results of the experiment are then presented and discussed, and it is demonstrated that Model A and Model B are both equally as useful in this pilot study as search based analysis.

6.1 Experimental Design

The evidence discovery system, as its name states, is an information discovery technique which is intended to explore the text data and enlighten the user by leading him/her through pathways in the data.

Information retrieval systems, such as search engines and search tools, are intended to retrieve portions of the data that are relevant to an input query. As discussed in Section 3.3.1, IR systems are typically measured by recall and precision. However, since the output of discovery systems are previously unknown or unrecognised, measuring the performance of discovery systems is more difficult than IR and IE systems (de Lin and Chalupsky, 2004). The measurement of recall when applied to the discovery domain, measures how many units of information that are supposed to be discovered have been discovered. Therefore, recall can only be measured when knowledge exists of what there was to be found. The concept of precision in the discovery domain, attempts to measure whether the results are at least plausible, or that something was found that was worth finding (de Lin and Chalupsky, 2004). Therefore, recall is the most appropriate measure to use to evaluate and compare an IR method with a discovery method, because both methods retrieve units of information that can be measured by recall.

One of the fundamental differences between IR and discovery is the requirement for initial background information. While a discovery system has the advantage of not requiring any initial insight into the data or input query terms, when evaluating the

two systems against each other, the users of the search method will require some initial background information to tell the user what to look for and to assist in choosing initial query terms.

In order to evaluate the analysis of data using each method, a user based test was designed based on the idea of a 'reading comprehension', whereby the user's knowledge and understanding of the text is tested. There are two main ways in which a user's knowledge can be tested: a set of questions (either multiple choice or essay style) could be posed which the user must answer after analysis of the text, or a general essay style question can be posed in which case the user must write in prose enough information to fully answer the initial question. There are a number of advantages and disadvantages for each question style which are considered and discussed below.

A list of questions may lead the user and guide his/her exploration or manner of searching and analysing, by giving the user little pieces of information in each question. This shortcoming can be met by only making the questions available to the user once they have analysed the text and by preventing the user from revisiting questions which have been answered. Even so, preceding questions may give the user insight into succeeding ones. The questions also limit the information to be extracted to a small subset of the relevant information, thereby skewing the recall results to only the information covered by the questions. A significant advantage of question answering is that it allows for ready and, to some extent, quantitative comparisons of answers between users. Each answer can be easily evaluated and the clarity provided by this style of testing is very advantagous.

A general essay style question requires the user to compose an answer based on his/her evaluation of which data is relevant and his/her understanding of the data. This style best achieves the goal of this particular test: namely to measure recall, to assess the amount of information the user has gained and extracted, and their understanding of the data. This question style does not lead the user to focus on certain portions of the data and does not limit the recall measure to a subset of information covered by a set of questions. Yet, an essay style test has disadvantages. It is difficult for users to know how much information to include in their composition and different users will have different ideas of what information is relevant in order to answer the question posed. In addition,

users will capture varying amounts of information depending on their individual reading and writing speed, diligence, effort, and performance.

While a multiple choice question testing mechanism would provide a way to quantitatively evaluate the models, the main aim of this study is to investigate the effectiveness and usability of the evidence discovery system for evidence discovery. Therefore, the 'essay style' testing mechanism was chosen, because it simulates how the tool would be used in a real scenario. Thus, a qualitative analysis will need to be done, to evaluate the performance of the evidence discovery system.

Quantitative methods typically depend on large samples in order to generalize with confidence from the sample to the population that it represents. However, according to Patton (2002), qualitative inquiry typically focuses on relatively small samples, even single cases, to permit enquiry into and understanding of an experience in depth. While one cannot generalise from single cases or small samples, one can learn a great deal from them (Patton, 2002). Given the qualitative nature of this study, a small samples size of four per group was chosen. This allows for each of the participants results to be analysed in depth to gain insight into how the user interacts with the system (Patton, 2002).

The amount of time allocated for analysis of the text and for capturing of results cannot be calculated or objectively decided. To measure the effect that the length of time allowed has on the performance of each of the methods, several studies would have to be conducted with several different time limits to compare the results. Therefore either a number of studies should be carried out or an estimate for the time allowance must be chosen. Because the scope of this work only allows for a preliminary experiment, a time estimate was made. The amount of time allowed needed to be chosen such that there is enough time for the user to discover the storyline, but not too much time that the user can just read the majority of the text. As there is a limited amount of relevant information to be retrieved, with enough time all the information can be found and extracted by all of the methods and their performance will eventually be similar. Thus a conservative estimate of one hour for text analysis and 30 min for capturing was chosen.

6.2 Methodology

Twelve random participants, each of which had some prior experience of search-based analysis methods, were randomly divided into three groups of four. Each group was then randomly assigned to an analysis method:

- Group 1 tested the evidence discovery system with the statistically based relation discovery model, Model A. Within the experiment this method as a whole will be referred to as **Model A**.
- Group 2 tested the evidence discovery system with the linguistically based relation discovery model, Model B. Within the experiment this method as a whole will be referred to as **Model B**.
- Group 3 tested a **search** method: TextPad (HeliosSoftwareSolutions, 2000) provides a search engine using UNIX-style regular expressions, with which the users could perform within document content searching.

For the reasons set out in Section 4.1, the novel entitled 'The Mysterious Affair at Styles' by Agatha Christie was used as a dataset for this research. The novel presents a mystery concerning a murder. The nature of Christie's writing is such that there are a number of red-herrings to confuse the reader, which present a challenge to the reader to solve the mystery of the crime. However, Christie typically reveals the manner in which the murder was committed and the identity of the murderer in a final paragraph or paragraphs. In order to make the novel more akin to a true dataset, this portion of the text was removed for all users.

Real evidence found in real data, in contrast to the fictitious 'evidence', can be unclear and uncertain and has a significantly greater noise component due to the data which is not regarded as having evidential value. For instance, only one small piece of evidence may be found in an entire hard drive, whereas the novel contains all of the evidence and facts required to solve the crime. It is therefore advantageous to use fictitious data for a comparative evaluation, because the evidence presented in the novel is clear and can be easily evaluated by reading the story. To assist the search-users, in choosing their initial query terms, all of the users are told that the text is a novel about a murder

mystery. None of the participants had read an Agatha Christie novel before the testing; in particular none had read 'The Mysterious Affair at Styles'.

Before commencing the test a demonstration of the software was given to the participants, using the book of 'Peter Pan' by James M. Barrie, after which participants were given 5 minutes to become acquainted with the tools themselves.

On commencement of the test, each user was allowed one hour to analyse and explore the test dataset to find 'leads' and 'evidence' regarding the crime in the test story and to formulate an 'hypothesis' as to who committed the crime. The objectives set out for the users were:

- Using the software provided, try to extract from the text the key elements that make up the plot such as the main characters and outline of the story.
- Who committed the crime and how?
- Each character, fact and event extracted awards you points weighted by their importance. The aim is to score as many points as possible.

After the permitted hour of analysis, each user was allowed a maximum of 30 minutes to write a synopsis or hypothesis of the story, which could be in the form of a few paragraphs describing the whole story, or a paragraph for each character describing who they are and how they fit into the story.

In order to compare and score the user's answers, a master marking sheet was drawn up containing all of the relevant facts (characters, relations and events) extracted from the story. Weights were then assigned to each item based on an assessment of its importance to the story and the difficulty in extracting it. The manner in which weights were assigned to various items of information from the text is set out in Appendix D. This assignment of weights to information is necessarily a subjective process as different persons may assign different levels of importance to elements of the plot or have differing opinions as to the difficulty involved in extracting certain information. However, each of the user's answers are marked and scored against the same master marking sheet such that any skewness in the scores was uniform.

After completion of the test, all the units of information each user had extracted were added up to determine (a) the total number of units of information extracted and

Table 6.1: Results from the preliminary experiment.

	Model A	**Model B**	**Search**
Mean (recall)	0.24975	0.290169	0.275948
Std dev (recall)	0.0821294	0.108424	0.138394
Mean (score)	0.320643	0.329450	0.266382
Std dev (score)	0.117804	0.111308	0.100209
No. data points	4	4	4

recorded by each user and (b) using the weights listed in the master marking sheet a weighted score intended to reflect the relative importance of the information extracted and recorded was calculated. Information extracted by a user but not recorded in their answer could obviously not be taken into account.

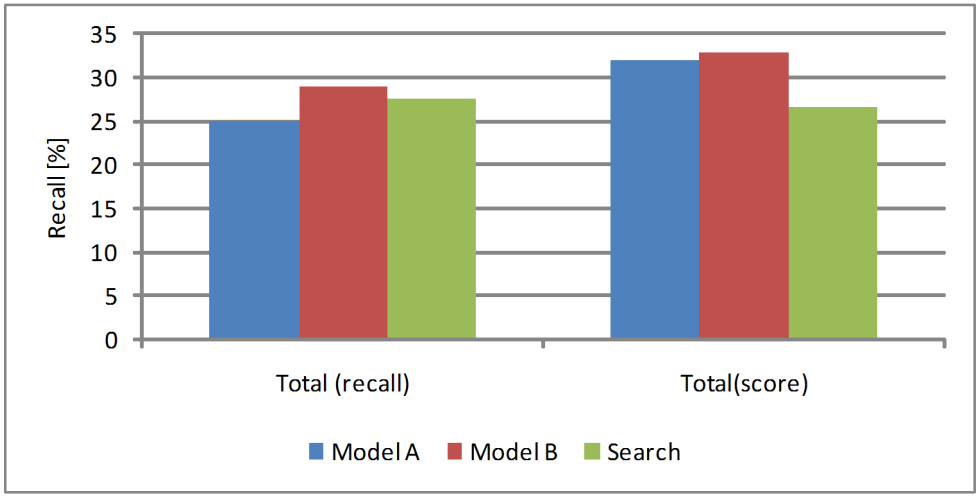
6.3 Results

The recall and score results of each group was characterised by its mean, standard deviation and number of data points. These values are shown in Table 6.1.

The recall of each group of participants shows the amount of information, or number of 'units of information' that were extracted as a percentage of the total number of units of information available for extraction. The score shows the relative importance of the information extracted by each group of participants expressed as a percentage. The recall and score totals for each group appear in Graph 6.1.

No significance can be attached to the variation between the recall and score means of the three groups due to the large standard deviations. The wide standard deviations indicate that there were significant differences in performance for each person. This is to be expected in a test of this nature and is the result of varying abilities, effort, personality types, experience, and aptitude of the participants, inconsistent reporting and recording of findings by participants and familiarity with the type of subject material.

The wide standard deviations within each group coupled with the closely set means across groups indicate that there is a high degree of overlap in the information extracted



Graph 6.1: Mean results for each of the models.

by the three groups. This also prevents any significance being attached to the difference in the means of the three groups.

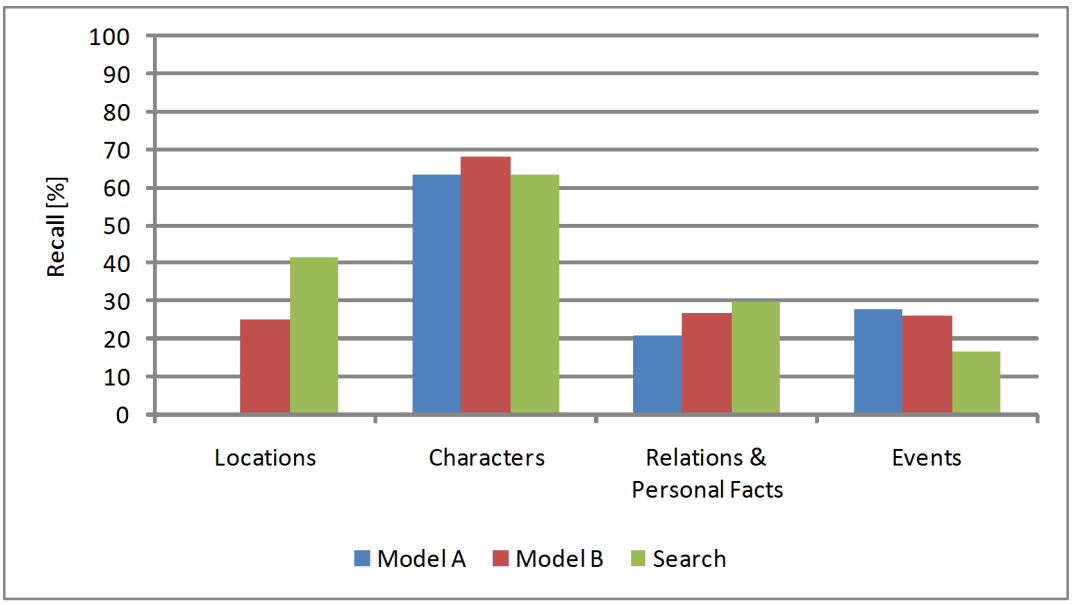
While it can be anticipated that there would be a greater dissimilarity between the results of each group as the time allowed for analysis is decreased, it is unlikely that the allowed analysis time of one hour was too long. Had the analysis time of one hour been too long one would have expected higher recall values for each group and lower standard deviations across the groups. In fact, no user achieved greater than 0.43 on the recall measure.

The rest of this section analyses and discusses the results of the experiment by each category in Section 6.3.1 followed by an analysis of the second objective 'who committed the crime and how' in Section 6.3.2.

6.3.1 Results by category

There are four categories of information that could be extracted from the novel: locations, characters, relations and personal facts, and events.

Locations are those entities that refer to the locations, towns or places. There are only three location references that are relevant to the story. These are presented in Table D.1. Characters refer to entities of type person, of which there are 22 significant characters in the story, presented in Table D.2. Types of relations to be extracted



Graph 6.2: Results for each model across the categories.

include familial relations (parents, children, wives), professional occupations (doctors, attorneys), friendships and acquaintances. Personal facts cover a broad range of useful pieces of information about the characters that are relevant to the storyline, such as 'Poirot is a Belgian detective' and 'Dr Bauerstein is an expert in poisons'. There are 45 relations and personal facts about the characters that are relevant to the storyline. The relations and personal facts that are required to be extracted are presented in Tables D.3-D.4. The final and largest category includes units of information describing what happened in the story, i.e. the storyline. There are 98 event units, which are presented in Tables D.5-D.9.

The mean score for each group in each of the four categories (expressed as a percentage) is shown in Graph 6.2. The scores for each of the groups within each category were generally similar. The only category with scores greater than 50% was Characters.

Locations

The category with the largest variation in scores appears to be Locations. In this category the Search method appears to perform much better than the other two methods. It should be borne in mind, however, that there are only three location references. Of

the participants using the Search method 50% found at most two locations and one participant did not find any. All of the relevant locations are also located together in the first few paragraphs of the text.

None of the participants using Model A found any locations. Investigation of Model A found that none of the locations occurred in the text often enough to meet the minimum frequency requirements imposed by Model A. The locations therefore would not have appeared in the graphical user interface with the result that the user would only have found location references if they happened to examine a sentence that is represented in the graph that contains a reference to the location. For example, the link for the nodes 'chemist' and 'strychnine' contains a sentence that states that the chemist's shop is in Styles St. Mary. However, because the location is not represented in the graph itself, the user is unlikely to place much value on the reference to the location. Therefore, the user is unlikely to extract references found in that manner.

Model B aims to compensate for the limitations of such frequency thresholds, by utilising NER. However, only 50% of those using Model B found at least one location. On examination it appeared that the poor accuracy of NE co-referencing had caused significant ambiguity. In this particular experiment it was found that the NE tagger could not properly differentiate and disambiguate the locations of the town Styles St. Mary and the residence Styles Court and they were tagged as a single entity. It was also found that one of the references to Styles St. Mary was allocated to the entity of Mary Cavendish. However, manual disambiguation of entities should be simple when the user considers the context of the references, by clicking on the node and reading the sentences displayed for the node.

The above indicates that when determining which extraction method would be most appropriate for the extraction of location information, one should carefully consider the nature of the text and the frequency with which location terms occur in the text. If the terms occur infrequently, they will fall below the threshold of Model A and will be excluded from the graph. Model B may then be the preferred method. However, if there are terms that are similar to other terms, some of them may be missed by Model B. While it is still difficult to find location terms using Search, in this case Search appears to have a better chance. For this type of information, a combination of methods may

achieve the best results. However, as there were only three location terms to be extracted in this experiment, these results are inconclusive.

Characters

All of the models performed significantly better in the Characters category than in any of the other categories. In this category the models performed very similarly, as illustrated in Graph 6.2.

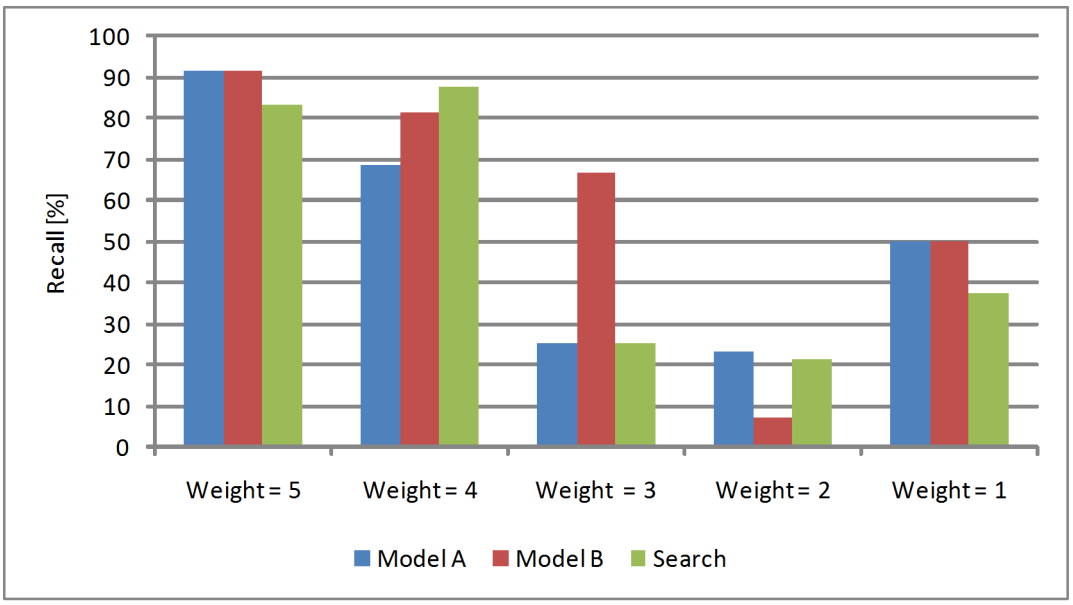
There are a number of reasons why characters were more easily found and extracted than other categories of information. In a novel, characters interact with each other which increases the likelihood of characters occurring near each other in the text. A user who finds one character is therefore likely to find another in addition. In a novel, the names of the characters are mentioned frequently and characters therefore score very well in a statistical based extraction method like Model A. The high frequency of characters and the likelihood of characters to appear in close proximity to each other in the text also assists the participants using the Search method, as the names are likely to appear in the text in close proximity to other terms that the user may be searching for. Once a character has been found, that character may also become a useful query term which assists the Search users to find other characters which interact with it. Characters, being named entities, are also assured of appearing in the Model B graphical user interface.

Since the objectives set for the users stated that they should extract the main characters from the story, the users probably put more effort into finding and extracting all of the characters and this information is probably also easier for a user to record.

It is therefore not surprising that users scored better in the characters category than in the other categories, regardless of the Model being used.

As described in the methodology, characters were assigned weights in the Master marking sheet based on their relative importance or contribution to the central plot of the novel. Characters can therefore be divided into weight groups according to the weight assigned to the character in the Master marking sheet. Graph 6.3 illustrates the percentage of characters extracted for each weight group, where the most important characters have a weight of 5 and the least important characters have a weight of 1.

The graph shows that the recall for ‘key’ characters, or most important characters,



Graph 6.3: Percentage of characters extracted for each weight grouping.

is higher than the other weight groups. One reason for this may be that key characters, being important to the novel's central plot, occur more frequently in the text than less important characters and therefore have a higher extraction rate. However, extraction of characters in the Weight 1 group does not fit this pattern. Examination of the characters in the Weight 1 group reveals that there are only two characters assigned to this group, namely Dorcas and Elizabeth Wells. Dorcas and Elizabeth are two of three housemaids in the story. They were assigned a weight of one, because they had little opportunity to be involved in the murder and thus, were not suspected. It is interesting to note that the character Dorcas had an exceptionally high extraction rate, an average of 75% across all of the participants, i.e. 75% of the participants extracted the character Dorcas. Since the assignment of weights is a subjective process, the high extraction rate is a strong indication that the weight chosen for Dorcas was too low. Assessment of the character Dorcas showed that she was frequently questioned by the detectives involved in the case, because she overheard some of the arguments between the victim and the suspects. Thus, re-evaluation of the weight assignment showed that the character Dorcas should have a weight of three. This would reduce the average extraction rate for the Weight 1 group to approximately 10%, which is inline with the expected extraction rate based

on importance and frequency. Changing the weight assigned to Dorcas to three had a negligible effect on the extraction performance for the Weight 3 group.

Graph 6.3 shows that Model A had a lower extraction rate than Search and Model B for characters in the Weight 4 group. It was found that the character Mr Hastings did not meet the threshold requirements of Model A to be included in the graph. Mr Hastings is the narrator in the book, and therefore references to him are most commonly and frequently referred to using "I". Unfortunately, unless NER and co-reference disambiguation is performed, it is very difficult for users of Model A to manually disambiguate the character Mr Hasting with the narrator "I", without referring to the full text. It was also found that the character Dr Wilkins, in the Weight 3 group, was excluded from the graph in Model A and therefore could not be extracted by any of the users of Model A. Dr Wilkins is referred to 19 times in the text and thus meets the frequency requirement for model A. However for inclusion in the graph, the node is required to be connected or related to another node and must meet the required support threshold for the link between the two nodes. Dr Wilkins did not meet the necessary support threshold for links to other nodes. It is interesting to note that only one of the Search users extracted the character Dr Wilkins, which can be attributed to the low extraction rate by Search users for the Weight 3 group.

Model B has a clear advantage over the other two models in extracting characters, due to its NER algorithm, which can be seen in its significantly better performance over both Model A and Search in the Weight 3 and Weight 1 character group. It is surprising to find that despite the automatic extraction and listing of NEs, Model B had an extraction rate of only 90% and 80% for the Weight 5 and Weight 4 groups respectively. Inspection of the results showed that three out of the four Model B users had extraction rates of 100% for both the Weight 5 and Weight 4 groups. The remaining user had extraction rates of 67% and 25% for these two groups respectively, which can be attributed to that user's evaluation of the importance of the characters to the storyline and/or the amount of effort he/she put into investigating and extracting characters and recording extracted results. In contrast the performance of Model A and Search for the Weight 5 and Weight 4 character groups was lower across multiple users and could not be attributed to just one user. The poor performance of Model B in the Weight 2 group was identified to be

due to errors in the NER algorithm, whereby five of the seven characters in the Weight 2 group were not identified by the NER algorithm and thus were not included in the NE list.

All three models are effective at extracting characters, but it is expected that Models A and B would have a clear advantage over Search where the text being explored is not a novel and characters do not occur near each other in the text or interact. In this case, unless Search users know the names of the characters they are looking for, Models A and B would perform better, because of their automatic extraction methods. Model B would probably perform better than Model A because of its NER algorithm and the need in Model A to satisfy both a frequency threshold and a node interaction threshold. If one wanted to find characters only and nothing else (no relationships or events) NER could be used alone, and in that way might be more efficient. A person therefore needs to have an understanding of the nature of the text to be explored and the type of information required to be extracted in order to know which model is most appropriate to be employed.

Relations and Personal Facts

Relations by definition represent relationships and interactions and are therefore described by the manner in which they link characters; or characters and occupations, interests, nationalities, and facts about the characters. Relations cannot be extracted or retrieved using a search method in the same way as a character or a location can. Characters and locations, being nouns, can be extracted independently of the rest of the information in the text. A relation can only be extracted by describing the relationship between a minimum of two pieces of information, for example: “Dr Bauerstein is a friend of Mary’s”, “John practiced as a barrister”. Therefore the extent to which one can extract relations is dependent on the extent to which one can extract relation independent information such as locations, characters, occupations and nationalities.

One possible explanation for relations having done poorly relative to characters is that users did not assign much importance to relations and did not expend much effort in extracting or recording them. That this might have been the case is supported by the fact that all three models did not perform well in this category.

This does not account, however, for the fact that search did better than either model in this category. Search probably outperformed the other models because the structure of a novel is such that when searching for characters one finds and reads the information about them, and thus one finds the relations. That this explanation is likely, is borne out by the fact that search does much better in relations than events, as is discussed below.

That Model A did not do well is perhaps expected as it does not focus on the links between NEs, it rather focuses on the links between frequent or important concepts.

If one wants to discover relations it may be most efficient to use a NER algorithm, followed by search, but this observation might only be true for novels and not for other forms of text.

Events

The events category is the largest category containing 98 'event-units' and is the only category in which the two evidence discovery models performed better than Search. Events are composed of one or more concepts and an associated action. This makes events harder to find and extract than characters or simple relations. The text contains numerous 'sideline' events that form sub-plots in the story, which make it difficult for users to find and identify the important events that solve the mystery. In the story the important events are widely distributed across the text and this makes it even more difficult to uncover these events using Search. Additionally, the Search users had to think of and choose query terms to search through the text to find events, using new query terms extracted from their results to make further progress. This makes it difficult for a Search user to find the events that are relevant to the main story line without reading the full text.

The evidence discovery users had the advantage of the graphical interface which in effect provided them with 'pre-populated search terms' in the form of the nodes in the graphs. Events may also be found more easily by the evidence discovery models because the concepts or characters in the events are connected by common nodes.

Model A connects important concepts and characters in a graph. Because events tend to revolve around important concepts and characters, these nodes have several

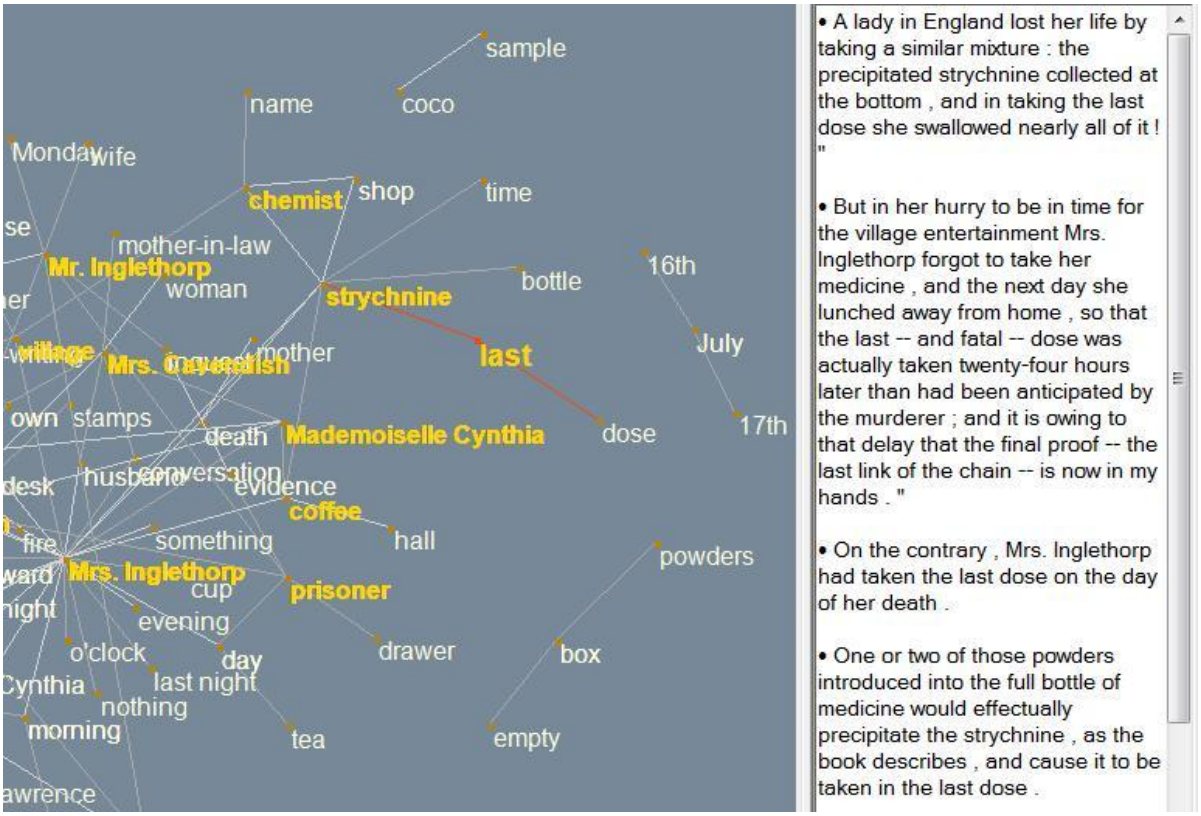


Figure 6.1: Analysis of events in Model A.

connections and can be easily found in the graph. Figure 6.1 shows an analysis of the links 'strychnine – last – dose' in the graph produced by Model A. The example in Figure 6.1 shows how the following two very important pieces of information can be found:

- "One or two of those powders introduced into the full bottle of medicine would effectually precipitate the strychnine, as the book describes, and cause it to be taken in the last dose."
- "But in her hurry to be in time for the village entertainment Mrs. Inglethorp forgot to take her medicine, and the next day she lunched away from home, so that the last – and fatal – dose was actually taken twenty-four hours later than had been anticipated by the murderer; and it is owing to that delay that the final proof – the last link of the chain – is now in my hands."

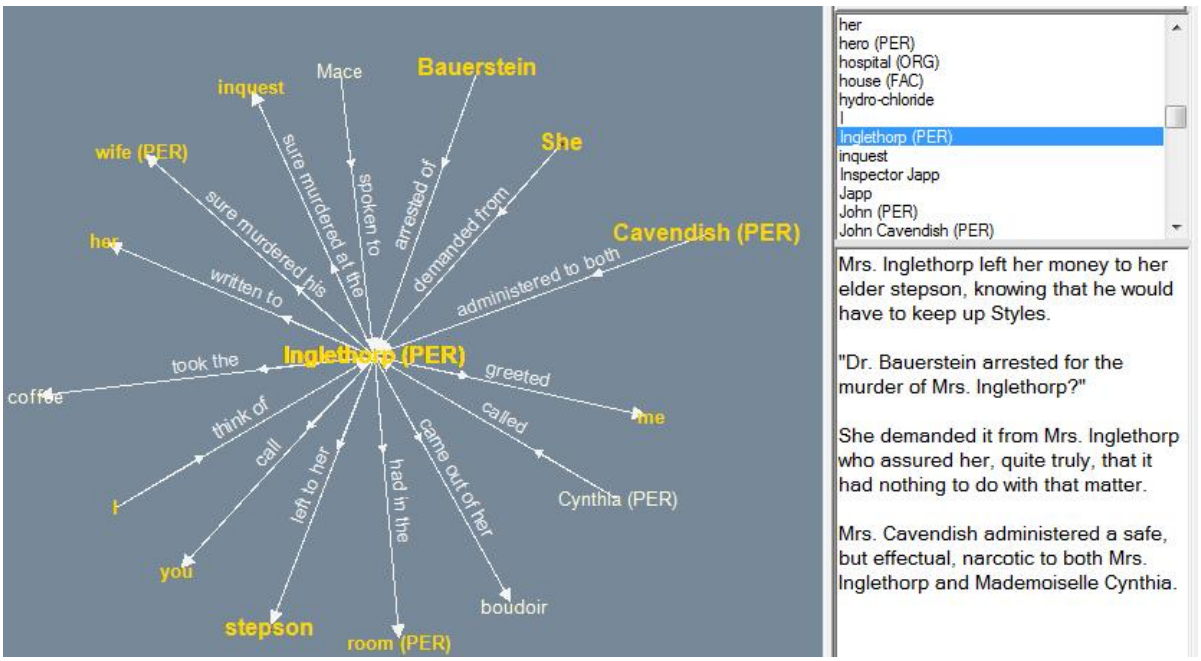


Figure 6.2: Analysis of events in Model B.

Model B is designed around NEs, and focuses on the events connected to the NEs. Similarly to Model A, events can be easily found by exploring the graphs. Figure 6.2 shows an analysis of the node 'Inglethorp' in the graph produced by Model B. The example in Figure 6.2 reveals an important event between 'Mrs Cavendish' and 'Mrs Inglethorp':

- "Mrs Cavendish administered a safe, but effectual, narcotic to both Mrs Inglethorp and Mademoiselle Cynthia."

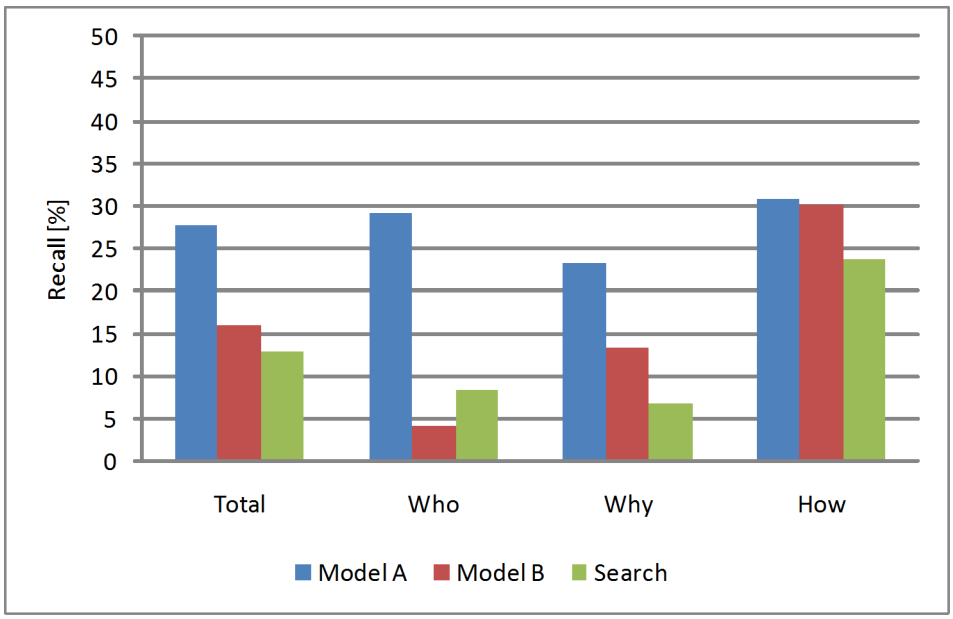
The evidence discovery models are best suited to the extraction of events and perform better than Search. In a small body of text one would expect a reading of the text or even perhaps searching to be adequate, equal to or perhaps even better than Models A or B, but this test shows that the longer and more information rich the text the better the evidence extraction models will perform in comparison to Search.

6.3.2 Who committed the crime and how

It is interesting to analyse how effective the three models were in solving the murder mystery in the text. It is important to remember that the final chapter of the novel was excluded from the text used in the experiment. This ensures that the conclusion which reveals the details of who committed the crime, how it was done and the murderer's motive was not given to the users. However, the nature of an Agatha Christie novel ensures that the reader is given all of the clues in the preceding chapters. In order to identify the criminal(s) and put together the pieces of how the crime was executed, the user is required to make deductions and decide which clues are fact and which are misleading 'red-herrings'.

In order to evaluate the second objective set for the users ('who committed the crime and how?'), each fact which contributes to the answer was put into one of three sets, namely Who, How, and Why. This required the user to have successfully extracted information from each of the four categories. Each of the Who, Why and How sets can accordingly be assigned a total absolute score calculated as the sum of the individual scores of the elements of that set, whether a location, character, relation or event. Graph 6.4 illustrates the mean score expressed as a percentage of the maximum possible score for each of the Who, Why and How sets for each method. The "Total" indicates each method's mean score across all three plot elements. It is evident from Graph 6.4 that in each set Model A performed best, followed by Model B and finally search. This mirrors the results for the Events category discussed above.

That the performance of each of the methods in extracting the plot mirrors their performance in extracting events is to be expected. Although a plot is comprised of locations, characters and relations as well as events, unless a user understands the events, they will have a list of disparate and discrete locations, characters and relations, and no sense of how these contribute to the whole. It is only through extracting events that a user can piece together a plot. It is therefore to be expected that the method that performs most well at extracting events would also perform best at enabling a user to extract a plot.



Graph 6.4: Analysis of results for Objective 2: 'who committed the crime and how?'.

Who

There were two people involved in the murder of Mrs Inglethorp. While five out of the twelve participants (approximately 40%, two Model A users, one Model B user, and two Search users) identified the main culprit (Miss Evelyn Howard), who planned and executed the murder, only one user was able to identify the accomplice, Mr Alfred Inglethorp. In order for the users to discover Miss Howard's involvement in the murder, the users needed to find the portion of the text that reveals the contents of a letter that Mrs Inglethorp found. The characters in the book had different ideas as to the contents of the letter, but it was found to be an unfinished letter of correspondence between the two culprits, who were planning the murder. To deduce who the accomplice was, the user needed to discover that Miss Howard and Alfred Inglethorp were 'distant relations' and perhaps conclude that despite the pretence of them hating each other, they in fact were planning on eloping together. The fact that the text contained several interpretations of the letter and that all three methods performed poorly at relation extraction accordingly made discovery of both murderers and an understanding of the relationship between them difficult.

Why

Approximately 50% of users for each model discovered that, on Mrs Inglethorp's death, Alfred Inglethorp would inherit her money. However, only two of the users, both using Model B, stated that the money from Mrs Inglethorp's will was the motive for the murder, despite the fact that neither of these users stated that Alfred Inglethorp was the murderer. Alfred's title to inherit the money coupled by Evelyn Howard being identified by most users as the murderer created confusion for the users, as these characters tried to hide their relationship by pretending to hate each other. Examination of the motive for the murder would have enabled the users to connect the inheritance of Mrs Inglethorp's money and the discovered fact that the murderer had an accomplice. In order for Miss Howard to benefit from the murder she had to have an agreement or arrangement with Alfred Inglethorp. Only one user, using Model A, deduced that a probable romantic relationship existed between Alfred Inglethorp and Miss Howard. This again demonstrates the importance of a method that performs well at extracting relations.

How

From Graph 6.4, it can be seen that Model B and Search perform considerably better in this set than in the other two sets. This is probably because the 'how' does not require an understanding of the relations and how all the pieces of the story fit together. The 'how' only considers events, which are directly concerned with how the murder was accomplished. Thus the performance of the models is determined by their ability to extract events from the text.

6.4 Feedback from the users

A discussion with the users confirmed that they thought the evidence discovery system could greatly aid an investigator in analysing digital data to reveal useful information and to discover evidence. A number of participants indicated they would like search functionality added to the evidence discovery system to a) be able to search for nodes within the graphs and b) to search for items in the source text that were found in the graph. The users indicated that they really liked and used the links or relations between

nodes. However, it could not be established whether or not the use of NER was useful and some users said they predominantly used the graphs, but didn't utilise the list of named entities.

6.5 Discussion

Although the preliminary experiment was not designed to evaluate the performance of Model A against Model B a discussion of the advantages and disadvantages of each can be used to identify how each of the models could be improved and give direction for future work.

Model A has the advantage of simple statistical algorithms that can be easily and accurately performed, whereas Model B is dependent on several linguistic algorithms which are each more error prone than the statistical methods and these errors are compounded in each step. Model B incorporates NER which, when it is done well, can be very useful to investigators. However, NER could be built into Model A, which could improve the usefulness of this model. Because the time complexity of the algorithms were not calculated, it is impossible to say which algorithm performs better with respect to time performance. However, this is an aspect that requires further investigation. Without taking the time complexity into account, the simplicity and accuracy of Model A makes it a more practical model for implementation.

The preliminary study conducted here indicates that the best choice of method employed for information extraction largely depends on the nature of the text being analysed and the type of information required to be extracted.

The evidence discovery models provide the user with an advantage that search methods cannot provide, regardless of the nature of the text, in that the user is provided with a graph of concepts, which could effectively be used as a description of the text and suggestions for search terms. By clicking on a node one has easy access into the text and all sentences containing that node. One also has a visual presentation of which nodes are linked to which other nodes. Even if the text were to ultimately prove to be unsuitable for either of the evidence discovery models, one might find the models useful when used in conjunction with search for this reason alone. A two phased approach using the evi-

dence discovery system first would prove useful to search users, because they would then approach their search exercise with the benefit of having background information about the text and the relationships between potential search query terms. The experimental results suggested that the evidence discovery users relied on the graph to direct whether or not a term is important, based on the assumption that if a term is important it will appear in the graph and if it is not in the graph it is not important. Where the model chosen is well paired with the type of data it is utilised on, this effect is very desirable. It may, however, be a drawback in cases where the models do not perform well and the user relies too heavily on the graph to determine the importance of concepts.

If one is looking to extract specific types of information a more directed and specific extraction method would probably be more effective. For example, if one were only interested in extracting characters and locations, an NER algorithm alone would suffice.

It was surprising to find that the Search method performed slightly better than the evidence discovery models in extracting relations. If one was required to extract relations only, a tailor made relation extraction algorithm utilising NER may be the most appropriate. Current information extraction algorithms work on the principle that relations can be extracted based on simple patterns using NEs and key words as anchors in a template, for example: "(Person) 'is a friend of' (Person)". However, currently these systems are only generic to a very limited extent and largely rely on the developer to manually build the templates to be used.

The evidence discovery models were most useful in extracting events. The search method was particularly poor at extracting event information and wherever the text contains any kind of narrative, a method designed to extract events will perform best and will outperform search. However, relation extraction is important to a proper understanding of narrative and the evidence discovery models did not perform particularly well in extracting relations. Future development could therefore include the integration of a domain or text specific relation extraction component, similar to the current template based information extraction algorithms.

6.6 Conclusion

This chapter evaluated the useability and usefulness of the results produced by the evidence discovery system. Because a full-scale quantitative assessment of the performance of the models is beyond the scope of this study, a preliminary study was devised with the aim of determining whether the evidence discovery system produces sensible, useful and useable results.

Random participants were asked to use the evidence discovery system to extract information from a text in a controlled environment. The information extracted was compared and assessed against information extracted by a control group of participants who extracted information from an identical text using content based search.

The results produced by each of the relation extraction models used within the evidence discovery framework were shown to be useful to the users in the experiment. It was concluded that the novel approach to text analysis for evidence discovery presented in this thesis is a viable and promising approach. The preliminary experiment showed that the results obtained from the evidence discovery system, using either of the relation extraction models, are sensible and useful. It was therefore concluded that the approach advocated in this thesis can be successfully applied to the analysis of textual data for digital forensics.

This dissertation is concluded in Chapter 7 and suggestions for future work are given.

Chapter 7

Conclusions

This chapter briefly summarises the findings of this work in Section 7.1 and considers future directions for related research resulting from these findings in Section 7.2.

7.1 Summary of Conclusions

The aim of this research was to investigate the adaptation and application of text mining methods to the analysis of textual data for the purpose of digital forensic investigation. First a study of the literature in digital forensics was conducted and presented in Chapter 2, and the requirements and needs for advancement in the analysis phase of a digital forensic investigation were identified. It was found that a major challenge exists for all investigators who wish to utilise digital evidence, in examining and organising the data to piece together events and facts of a crime. It was established that the difficulty with finding relevant information quickly using the current tools and methods is that these tools rely very heavily on background knowledge for query terms and do not fully utilise the content of the data. A review of the literature showed that, to make the investigation process more efficient, tools need to be developed that will reduce the quantity of data which requires analysis, aid the analysts' exploration of the data, and enhance the intelligibility of the presentation of the data.

Chapter 3 presented a survey of the significant literature in text mining. The literature showed that at present, automatic information extraction systems require large

amounts of domain knowledge to be embedded into the system to make it usable for a new domain. It was identified that domain independent information extraction systems using un-annotated text still face many challenges. The reason for this was found to be that the performance of current language analysis tools are inadequate, causing system designs to require significant user interaction and making the design of a new system a time-intensive task. The literature showed that more sophisticated language analysis tools are needed to get better and more usable results from text mining. Visual data exploration was identified to be particularly useful when little is known about the text data or when the exploration goals are vague. It was found that the visual data exploration process can be useful in assisting the user to generate, develop and validate hypotheses, as well as to explore and gain insight into the data.

This dissertation hypothesized that information extraction techniques combined with visual exploration techniques can assist in identifying suspects and events, and the relations between these entities. These techniques could assist an investigator to: piece together the story surrounding a crime, create hypothesises or potential leads for further investigation, or identify pieces of data which could form useful supporting evidence in a trial.

Chapter 4 proposed a novel framework in which to perform evidence discovery in an attempt to meet the challenges identified. The framework combines information extraction techniques with visual exploration techniques to provide a novel approach to performing evidence discovery, in the form of an evidence discovery system. By utilising unrestricted, unsupervised information extraction techniques, the investigator does not require input queries or keywords for searching, thus enabling the investigator to analyse portions of the data that may not have been identified by keyword searches.

The evidence discovery system produces text graphs of the most important concepts and associations extracted from the full text to establish ties between the concepts and provide an overview and general representation of the text. Through an interactive visual interface the investigator can explore the data to identify suspects, events and the relations between suspects. This assists the investigator to piece together the story surrounding the crime, create hypotheses for potential leads for further investigation, or identify pieces of data which could form useful supporting evidence in a trial.

Chapter 5 presented two models for performing the relation extraction process of the evidence discovery framework. Model A takes a statistical approach to discovering relations based on co-occurrences of complex concepts. Model B utilises a linguistic approach using NE extraction and IE patterns. By working within a framework, individual components can be worked on and improved in a comparable manner.

A preliminary study was performed to assess the usefulness of a text mining research approach to digital forensics as against the traditional IR approach. The experimental design, methodology, and results were presented and discussed in Chapter 6. The results produced by each of the relation extraction models used within the evidence discovery framework were shown to be useful to the users in the experiment. The novel approach to text analysis for evidence discovery presented in this thesis is therefore a viable and promising approach. The preliminary experiment showed that the results obtained from the evidence discovery system, using either of the relation extraction models, are sensible and useful.

7.2 Future Work

The relative newness of the field of digital forensics means that the scope for future research is wide open. There are many fields that are producing interesting research, with much promise for cross-disciplinary application, such as link analysis, social network analysis, temporal text mining, and temporal text analysis. A few suggestions for future research to advance the analysis of digital data for evidence discovery that have been inspired by this research are given below.

7.2.1 Performing a case study on real data

In order to gain a clear insight into the usefulness of the evidence discovery framework, a case study should be performed on a real data set. This would identify any problems of practical implementation of the framework and provide the best insight as to how the framework and its components should be worked on and improved. A case study of this nature would clarify the needs and requirements of the users to improve their analysis of the data, in order to make the investigation process more accurate and efficient.

7.2.2 Theme extraction

An investigation of theme extraction could produce a promising extension to the evidence discovery framework. Once relations have been discovered, extracted, and linked to form graphs, graph analysis techniques could be applied to recognize themes among the relations. Themes may possibly present themselves in the graphs as long threads of connected nodes, or circular or other patterns of connected nodes. Extracted themes presented to the user could assist the user to focus on or rule out certain portions of the text to be analysed.

7.2.3 Information visualisation

Although the evidence discovery framework utilised an information visualisation component, it was not one of the main areas focused on in this research. There is much room for improvement in choosing or developing an optimal visualisation technique for the evidence discovery framework. Very large graphs with hundreds of thousand of nodes impact poorly on the performance and time complexity of visualisation and a user may not be able to discern the different nodes of the graph. Thus large graphs are often reduced or clustered before traditional layout algorithms are used. This is an important aspect of the framework to be investigated and would need to be addressed when looking at the scalability of the framework.

7.2.4 Addressing the needs of linguistic tools

In this research, it was found that systems using un-annotated text still face many challenges, because the performance of current language analysis tools is still poor. The main areas holding back the expansion, portability, and performance of current information extraction techniques were identified to be the language analysers which perform extended named entity tagging, dependency analysis, and co-referencing analysers. Many IE algorithms developed depend on the use of entities as anchors to recognise semantic units; therefore co-referencing (the merging of information referring to the same entity or event as well as the disambiguation of incompletely defied entities or events) plays an important role.

One of the ways in which the co-referencing of NEs could be addressed in the evidence discovery system could be to incorporate 'interactive NE co-referencing', whereby the user is able to edit NEs and other nodes to manually correct references previously assigned by the system.

7.2.5 Best of both worlds: discovery and search

The internet and search engines have brought search technology to all computer users. Most users are familiar and well practiced with search technology, which makes them more efficient utilising search tools as opposed to the newer and unfamiliar discovery tools.

It is thought that a best of 'both worlds approach' could easily be achieved by combining search-based analysis methods with discovery methods. Search functionality could be easily incorporated in the discovery system to enable the user to search for nodes within the text graphs and to search for items in the source text that were found in the graph.

Bibliography

Abraham, T. (2006). Event sequence mining to develop profiles for computer forensic investigation purposes. In *ACSW Frontiers '06: Proceedings of the 2006 Australasian workshops on Grid computing and e-research*, (pp. 145–153). Darlinghurst, Australia, Australia: Australian Computer Society, Inc.

AccessData (2008). Forensic tool kit 2.0.
URL `http://www.accessdata.com/forensictoolkit.html`

Agrawal, R., Imielinski, T. and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of ACM SIGOD Conference on Management of Data*, (pp. 207–216). Washington DC, USA.

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. In J. B. Bocca, M. Jarke, and C. Zaniolo (Eds.) *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB*, (pp. 487–499). Santiago, Chile: Morgan Kaufmann.
URL `http://citeseer.ist.psu.edu/agrawal94fast.html`

Asahara, M. and Matsumoto, Y. (2003). Japanese named entity extraction with redundant morphological analysis. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, (pp. 8–15). Morristown, NJ, USA: Association for Computational Linguistics.

Azé, J., Roche, M., Kodratoff, Y. and Sebag, M. (2005). Preference learning in terminol-

ogy extraction: A roc-based approach. In *Proceeedings of Applied Stochastic Models and Data Analysis*, (pp. 209–219).

Bardsley, M., Bel, R. and Lohr, D. (2005). Btk kansas serial killer - full btk story. World Wide Web. Last accessed 26/08/2008.
URL `http://www.crimelibrary.com/serial_killers/unsolved/btk/index_1.html`

Battista, G. D., Tollis, I. G., Eades, P. and Tamassia, R. (1998). *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall.

Bayardo, R. J. (1998). Efficiently mining long patterns from databases. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, (pp. 85–93). New York, NY, USA: ACM.

Beebe, J. G., Nicole Lang; Clark (2007). Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results. In *Digital Investigation*, vol. 4, (pp. 49–54). Elsevier Ltd.

Bertault, F. (1999). A force-directed algorithm that preserves edge crossing properties. In *Proceedings of the 7th International Symposium on Graph Drawing*, (pp. 351–358). London, UK: Springer-Verlag.

Bikel, D. M., Schwartz, R. and Weischedel, R. M. (1999). An algorithm that learns what's in a name. *Machine Learning*, *34*(1-3), 211–231.

Borthwick, A., Sterling, J., Agichtein, E. and Grishman, R. (1998). Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In *Proceedings of the 6th Workshop on Very Large Corpora*, (pp. 152–160).

Boyack, K. W., Wylie, B. N. and Davidson, G. S. (2002). Domain visualization using vxinsight for science and technology management. *Journal of the American Society for Information Science and Technology*, *53*(9), 764–774.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the Workshop on Speech and Natural Language*, (pp. 112–116). Morristown, NJ, USA: Association for Computational Linguistics.

Brown, L. (Ed.) (1993). *The New Shorter Oxford English Dictionary on Historical Principals*, vol. 1. Oxford University Press.

Califf, M. E. and Mooney, R. J. (2003). Bottom-up relational learning of pattern matching rules for information extraction. *Journal of Machine Learning Research*, *4*, 177–210.

Card, S. K., Mackinlay, J. D. and Shneiderman, B. (1999). *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann.

Carenini, G., Ng, R. T. and Zwart, E. (2005). Extracting knowledge from evaluative text. In *Proceedings of the 3rd International Conference on Knowledge Capture*, (pp. 11–18). New York, NY, USA: ACM.

Carrier, B. (2008). The sleuth kit v2.5. Last accessed 18/07/2008.
URL `http://www.sleuthkit.org/`

Carriere, J. and Kazman, R. (1995). Research report: Interacting with huge hierarchies: beyond cone trees. In *Proceedings of the 1995 IEEE Symposium on Information Visualization*, (p. 74). Washington, DC, USA: IEEE Computer Society.

Casey, E. (2004). *Digital Evidence and Computer Crime*. Academic Press.

Cavnar, W. B. and Trenkle, J. M. (1994). N-gram based text categorization. In *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval*, (pp. 161–175). Las Vegas, USA.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. and Wirth, R. (2008). Crisp-dm 1.0: Step-by-step data mining guide.
URL `http://www.crisp-dm.org/index.htm`

Charniak, E. (2005). The charniak parser (nlparser).
ftp://ftp.cs.brown.edu/pub/nlparser/.

Chen, C. (2004). *Information Visualization: Beyond the Horizon*. Springer.

Chen, H., Chung, W., Xu, J. J., Wang, G., Qin, Y. and Chau, M. (2004). Crime data mining: A general framework and some examples. *Computer*, *37*(4), 50–56.

Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics*, *16*(1), 22–29.

Churcher, N., Irwin, W. and Cook, C. (2004). Inhomogeneous force-directed layout algorithms in the visualisation pipeline: from layouts to visualisations. In *Proceedings of the 2004 Australasian Symposium on Information Visualisation*, (pp. 43–51). Darlinghurst, Australia, Australia: Australian Computer Society, Inc.

Collins, M. (2001). Ranking algorithms for named-entity extraction: boosting and the voted perceptron. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, (pp. 489–496). Morristown, NJ, USA: Association for Computational Linguistics.

Cruz, I. F. and Twarog, J. P. (1995). 3d graph drawing with simulated annealing. In *Proceedings of the Symposium on Graph Drawing*, (pp. 162–65). London, UK: Springer-Verlag.

Daelemans, W. and Osborne, M. (Eds.) (2003). *Seventh Conference on Natural Language Learning*. Association for Computational Linguistics, Morgan Kaufman Publishers.

Daille, B. (1996). Study and implementation of combined techniques for automatic extraction of terminology. In J. Klavans, and P. Resnik (Eds.) *The Balancing Act: Combining Symbolic and Statistical Approaches to Language*, (pp. 49–66). Cambridge, Massachusetts: The MIT Press.
URL `http://citeseer.ist.psu.edu/587322.html`

Daille, B., Gaussier, E. and Langé, J.-M. (1998). An evaluation of statistical scores for word association. In J. Ginzburg, Z. Khasidashvili, C. Vogel, J.-J. Lvy, and E. Valldnvi (Eds.) *The Tblisi Symposium on Logic, Language and Computation: Selected Papers*, (pp. 177–188). CSLI Publications.

Davidson, R. and Harel, D. (1996). Drawing graphs nicely using simulated annealing. *ACM Transactions on Graphics*, *15*(4), 301–331.

de Lin, S. and Chalupsky, H. (2004). Issues of verification for unsupervised discovery systems. In *Workshop on Link Analysis and Group Detection*.

de Vel, O., Anderson, A., Corney, M. and Mohay, G. (2001). Mining e-mail content for author identification forensics. *SIGMOD Record*, *30*, 55–64.

de Waal, A., Venter, J. and Barnard, E. (2008). Applying topic modelling on forensic data: A case study. In *Proceedings of the 4th Annual IFIP WG 11.9 International Conference on Digital Forensics*. Kyoto, Japan, Springer.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K. and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*, 391–407.

Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S. and Weischedel, R. (2004). The automatic content extraction (ace) program - tasks, data, and evaluation. *Proceedings of Language Resources and Evaluation Conference*, (pp. 837–840).

Dörre, J., Gerstl, P. and Seiffert, R. (1999). Text mining: finding nuggets in mountains of textual data. In *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 398–401). New York, NY, USA: ACM.

Doyle, L. B. (1961). Semantic road maps for literature searchers. *Journal of the ACM*, *8*(4), 553–578.

Dozier, C. and Jackson, P. (2005). Mining text for expert witnesses. *IEEE Software*, *22*(3), 94–100.

Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S. and Harshman, R. (1988). Using latent semantic analysis to improve access to textual information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (pp. 281–285). New York, NY, USA: ACM.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, *19*, 61–74.

Eades, P. A. (1984). A heuristic for graph drawing. In *Congressus Numerantium*, vol. 42, (pp. 149–160).

Fan, W., Wallace, L., Rich, S. and Zhang, Z. (2006). Tapping the power of text mining. *Communications of the ACM*, *49*(9), 76–82.

Farmer, D. and Venema, W. (2007). The coroner's toolkit v1.18.
URL `http://www.porcupine.org/forensics/tct.html`

Fawcett, T. and Provost, F. (1997). Adaptive fraud detection. *Data Mining and Knowledge Discovery*, *1*(3), 291–316.

Feldman, R., Fresko, M., Kinar, Y., Lindell, Y., Liphstat, O., Rajman, M., Schler, Y. and Zamir, O. (1998). Text mining at the term level. In *Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery*, (pp. 65–73). London, UK: Springer-Verlag.

Feldman, R. and Hirsh, H. (1996). Mining associations in text in the presence of background knowledge. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, (pp. 343–346).

Fellbaum, C. (Ed.) (1998). *WordNet. An Electronic Lexical Database*. MIT Press.

Fortuna, B., Mladenić, D. and Grobelnik, M. (2005). Visualization of text document corpus. *Informatica Journal*, *29*(4), 497–502.

Fox, K. L., Frieder, O., Knepper, M. M. and Snowberg, E. J. (1999). Sentinel: a multiple engine information retrieval and visualization system. *Journal of the American Society for Information Science*, *50*(7), 616–625.

FOXNews (2005). Cops make arrest in btk probe. FOXNews.com. Last accessed 1/08/2008.
URL `http://www.foxnews.com/story/0,2933,148794,00.html`

Frawley, W. J., Shapiro, P. G. and Matheus, C. J. (1992). Knowledge discovery in databases - an overview. *AI Magazine*, *13*, 57–70.
URL `http://citeseer.ist.psu.edu/frawley92knowledge.html`

Freitag, D. and McCallum, A. (2000). Information extraction with HMM structures learned by stochastic optimization. In *Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence*, (pp. 584–589). AAAI Press / The MIT Press.

Frick, A., Ludwig, A. and Mehldau, H. (1995). A fast adaptive layout algorithm for undirected graphs. In *Proceedings of the DIMACS International Workshop on Graph Drawing*, (pp. 388–403). London, UK: Springer-Verlag.

Fruchterman, T. M. J. and Reingold, E. M. (1991). Graph drawing by force-directed placement. *Software - Practice & Experience*, *21*(11), 1129–1164.

Gale, W. A. and Church, K. W. (1991). Concordances for parallel text. In *Proceedings of the Seventh Annual Conference of the UW Center for the New OED and Text Research, Using Corpora*, (pp. 40–62).

Greenwood, M. A., Stevenson, M., Guo, Y., Harkema, H. and Roberts, A. (2005). Automatically acquiring a linguistically motivated genic interaction extraction system. In *Proceedings of the 4th Learning Language in Logic Workshop*.

Grefenstette, G. and Tapanainen, P. (1994). What is a word, what is a sentence? Problems of tokenization. Technical report, Rank Xerox Research Centre Grenoble Laboratory.

Grishman, R. (2007). Jet (java extraction toolkit). Last accessed 10/04/2007.
URL `http://cs.nyu.edu`

Grishman, R. and Sundheim, B. (1996). Message understanding conference-6: a brief history. In *Proceedings of the 16th conference on Computational linguistics*, (pp. 466–471). Morristown, NJ, USA: Association for Computational Linguistics.

Grobelnik, M. and Mladenić, D. (2004). Visualization of news articles. In *SIKDD 2004 at multiconference IS*, (pp. 46–52). Ljubljana, Slovenia.

Guernsey, L. (2003). Digging for nuggets of wisdom. The New York Times. Last accessed 17/6/2008.
URL \protect\kern+.1667em\relaxhttp://query.nytimes.com/gst/fullpage.html?res=950CE5DD173EF935A25753C1A9659C8B63\&sec=\&spon=\&pagewanted=1

GuidanceSoftware (2002). Encase forensic v6. Last accessed 25/08/2008.
URL http://www.guidancesoftware.com/products/ef_index.aspx

Hachey, B. and Grover, C. (2005). Sequence modelling for sentence classification in a legal summarisation system. In *Proceedings of the 2005 ACM symposium on Applied computing*, (pp. 292–296). New York, NY, USA: ACM.

Han, J., Pei, J., Yin, Y. and Mao, R. (2004). Mining frequent patterns without candidate generation. *Data Mining and Knowledge Discovery*, *8*, 53–87.

Hasegawa, T., Sekine, S. and Grishman, R. (2004). Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, (p. 415). Morristown, NJ, USA: Association for Computational Linguistics.

Havre, S., Hetzler, E., Perrine, K., Jurrus, E. and Miller, N. (2001). Interactive visualization of multiple query results. In *Proceedings of the IEEE Symposium on Information Visualization*, (p. 105). Washington, DC, USA: IEEE Computer Society.

Hearst, M. (2003). What is text mining? World Wide Web.
URL http://people.ischool.berkeley.edu/~hearst/text-mining.html

Hearst, M. A. and Karadi, C. (1997). Cat-a-cone: an interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. *SIGIR Forum*, *31*(SI), 246–255.

HeliosSoftwareSolutions (2000). Textpad 4.4.0. www.TextPad.com.

Herman, I., Delest, M. and Melancon, G. (1998). Tree visualisation and navigation clues for information visualisation. Technical Report INS-R9806, Centre for Mathematics and Computer Science, Amsterdam, The Netherlands.

Herman, I., Melanon, G. and Marshall, M. S. (2000). Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, *6*(1), 24–43.

Hershkop, S. (2006). *Behavior-based email analysis with application to spam detection*. Ph.D. thesis, Columbia University, New York, NY, USA.

Honkela, T. (1997). *Self-Organizing Maps in Natural Language Processing*. Ph.D. thesis, Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland.

Hotho, A., Nrnberger, A. and Paa, G. (2005). A brief survey of text mining. *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology*, *20*(1), 19–62.

Infogistics (2001). Nlprocessor - text analysis toolkit. Last accessed 19/11/2008. URL `http://www.infogistics.com/textanalysis.html`

Johnson, B. and Shneiderman, B. (1991). Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the 2nd IEEE Conference on Visualization*, (pp. 284–291). Los Alamitos, CA, USA: IEEE Computer Society Press.

Jungnickel, D. (2004). *Graphs, Networks and Algorithms*. Springer, 2nd ed.

Jurafsky, D. and Martin, J. H. (2008). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson Prentice Hall, 2nd ed.

Keim, D. A. (2002). Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, *8*(1), 1–8.

Kodratoff, Y. (1999). Knowledge discovery in texts: A definition, and applications. In *Proceedings of the 11th International Symposium on Foundations of Intelligent Systems*, (pp. 16–29). London, UK: Springer-Verlag.

Kohonen, T., Kaski, S., Lagus, K., Salojarvi, J., Paatero, V. and Saarela, A. (2000). Self organization of a massive document collection. *IEEE Transactions on Neural Networks*, *11*(3), 574 – 585.

Lamping, J. and Rao, R. (1996). The hyperbolic browser: A focus+context technique for visualizing large hierarchies. *Journal of Visual Languages & Computing*, *7*(1), 33–55.

LDC (2005). *ACE (Automatic Content Extraction) English Annotation Guidelines for Entities*. Linguistic Data Consortium, University of Pennsylvania. V 5.6.1 2005.05.23. URL `http://www.ldc.upenn.edu/Projects/ACE/`

Lebert, M. (2008). *Project Gutenberg (1971-2008)*. Project Gutenberg. EText-No. 27045. URL `http://www.gutenberg.org/etext/27045`

Leskovec, J. and Grobelnik, M. (2004). Learning sub-structures of document semantic graphs for document summarization. In *Workshop on Link Analysis and Group Detection*, (pp. 133–138).

Lin, D. I. and Kedem, Z. (1998). Pincer-search: A new algorithm for discovering the maximum frequent set. In *Proceedings of the 6th International Conference on Extending Database Technology*, (pp. 105–119).

Lin, X., Soergel, D. and Marchionini, G. (1991). A self-organizing semantic map for information retrieval. In *Proceedings of the 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, (pp. 262–269). New York, NY, USA: ACM.

Lochbaum, K. E. and Streeter, L. A. (1989). Comparing and combining the effectiveness of latent semantic indexing and the ordinary vector space model for information retrieval. *Information Processing and Management: an International Journal*, *25*(6), 665–676.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N. and Watkins, C. (2002). Text classification using string kernels. *The Journal of Machine Learning Research*, *2*, 419–444.

Louis, A., Waal, A. D. and Venter, C. (2006). Named entity recognition in a south african context. In *Proceedings of the 2006 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries*, (pp. 170–179). Somerset West, South Africa.

Manning, C. D., Raghavan, P. and Schtze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Manning, C. D. and Schütze, H. (2001). *Foundations of Statistical Natural Language Processing.*. Cambridge, MA: MIT Press.

Marcus, M. P., Santorini, B. and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, *19*(2), 313–330.

Martí, R. and Laguna, M. (2003). Heuristics and meta-heuristics for 2-layer straight line crossing minimization. *Discrete Applied Mathematics*, *127*(3), 665–678.

McCallum, A. and Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the 7th conference on Natural language learning at HLT-NAACL 2003*, (pp. 188–191). Morristown, NJ, USA: Association for Computational Linguistics.

Mei, Q. and Zhai, C. (2005). Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, (pp. 198–207). New York, NY, USA: ACM.

Merkl, D. (1998). Text classification with self-organizing maps: Some lessons learned. *Neurocomputing*, *21*(1-3), 61–77.
URL `http://dx.doi.org/10.1016/S0925-2312(98)00032-0`

Meyers, A., Grishman, R., Kosaka, M. and Zhao, S. (2001). Covering treebanks with glarf. In *Proceedings of the ACL 2001 Workshop on Human Language Technology and Knowledge Management*, (pp. 51–58). Morristown, NJ, USA: Association for Computational Linguistics.

Muc (1998). *Proceedings of the Seventh Message Understanding Conference*. San Francisco, CA: Morgan Kaufmann.

Munzer, T. (1997). H3: Laying out large directed graphs in 3d hyperbolic space. In *Proceedings of the 1997 IEEE Symposium on Information Visualization*, (pp. 2–10). IEEE CS Press.

Munzner, T. (1998). Drawing large graphs with h3viewer and site manager. In *Proceedings of the Symposium on Graph Drawing*, (pp. 384–393). Springer-Verlag.

Nadeau, David, Sekine and Satoshi (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes*, *30*(1), 3–26.

Palmer, G. (2001). A road map for digital forensic research. Technical report, Report From the First Digital Forensic Research Workshop, Utica, New York.
URL `http://www.dfrws.org/archives.shtml`

Patton, M. Q. (2002). *Qualitative Research and Evaluation Methods*. SAGE, 3rd ed.

Patwardhan, S. and Riloff, E. (2006). Learning domain specific information extraction patterns from the web. In *Proceedings of the Workshop on Information Extraction beyond the Document*.

Purchase, H. C. (1997). Which aesthetic has the greatest effect on human understanding? In *Proceedings of the 5th International Symposium on Graph Drawing*, (pp. 248–261). London, UK: Springer-Verlag.

Raghavan, P. and Tsaparas, P. (2002). Mining significant associations in large scale text corpora. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, (p. 402). Washington, DC, USA: IEEE Computer Society.

Rajaraman, K. and Tan, A.-H. (2001). Topic detection, tracking, and trend analysis using self-organizing neural networks. In *Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, (pp. 102–107). London, UK: Springer-Verlag.

Rajman, M. and Besancon, R. (1998). Text mining - knowledge extraction from unstructured textual data. In *Proceedings of the sixth Conference of International Federation of Classification Societies*, (pp. 473–480). Rome.

Reingold, E. and Tilford, J. (1981). Tidier drawings of trees. *IEEE Transactions on Software Engineering*, *SE-7*(2), 223– 228.

Rekimoto, J. and Green, M. (1993). The information cube: Using transparency in 3d information visualisation. In *Proceedings of the 3rd Annual Workshop on Information Technologies & Systems*.

Rijsbergen, C. J. V. (1986). A non-classical logic for information retrieval. *The Computer Journal*, *29*(6), 481–485.

Riloff, E. (1996). Automatically generating extraction patterns from untagged text. In *Proceedings of the 13th National Conference on Artificial Intelligence*, (pp. 1044–1049).

Robertson, G. G., Mackinlay, J. D. and Card, S. K. (1991). Cone trees: Animated 3d visualizations of hierarchical information. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (pp. 189–194). New York, NY, USA: ACM.

Robertson, S. E. (1977). The probability ranking principle. *Journal of Documentation*, *33*, 294–304.

Roche, M., Azé, J., Matte-Tailliez, O. and Kodratoff, Y. (2004). Mining texts by association rules discovery in a technical corpus. In *New Trends in Intelligent Information Processing and Web Mining*.

Roth, D. and van den Bosch, A. (Eds.) (2002). *Sixth Workshop on Computational Language Learning (CoNLL-2002)*. Association for Computational Linguistics, Morgan Kaufman Publishers.

Salton, G. and McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill.

Salton, G., Wong, A. and Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620.

Sebastiani, F. (1999). A tutorial on automated text categorisation. In *Proceedings of the 1st Argentinian Symposium on Artificial Intelligence*, (pp. 7–35). Buenos Aires, AR.

Sekine, S. (2006). On-demand information extraction. In *Proceedings of the COLING/ACL on Main conference poster sessions*, (pp. 731–738). Morristown, NJ, USA: Association for Computational Linguistics.

Sekine, S. and Eriguchi, Y. (2000). Japanese named entity extraction evaluation: analysis of results. In *Proceedings of the 18th conference on Computational linguistics*, (pp. 1106–1110). Morristown, NJ, USA: Association for Computational Linguistics.

Shinyama, Y. and Sekine, S. (2006). Preemptive information extraction using unrestricted relation discovery. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, (pp. 304–311). Morristown, NJ, USA: Association for Computational Linguistics.

Sindre, G., Gulla, B. and Jokstad, H. (1993). Onion graphs: Aesthetics and layout. In *Proceedings of the 1993 IEEE Symposium on Visual Languages*, vol. 24, (pp. 287–291).

Smadja, F., McKeown, K. R. and Hatzivassiloglou, V. (1996). Translating collocations for bilingual lexicons: a statistical approach. *Computational Linguistics*, *22*(1), 1–38.

Small, H. (1999). Visualizing science by citation mapping. *Journal of the American Society for Information Science*, *50*(9), 799–813.

Smyth, P. and Goodman, R. M. (1992). An information theoretic approach to rule induction from databases. *IEEE Transactions on Knowledge and Data Engineering*, *4*(4), 301–316.

Spoerri, A. (1995). *InfoCrystal: a Visual Tool for Information Retrieval*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.
URL `http://www.scils.rutgers.edu/~aspoerri/InfoCrystal/InfoCrystal.htm`

Steinbach, M., Ertz, L. and Kumar, V. (2003). The challenges of clustering high-dimensional data. In *New Vistas in Statistical Physics: Applications in Econophysics, Bioinformatics, and Pattern Recognition*. Springer-Verlag.

Steinbach, M., Karypis, G. and Kumar, V. (2000). A comparison of document clustering techniques. In *KDD Workshop on Text Mining*.

Stolfo, S. J., Hershkop, S., Hu, C.-W., Li, W.-J., Nimeskern, O. and Wang, K. (2006). Behavior-based modeling and its application to email analysis. *ACM Transactions on Internet Technology*, *6*(2), 187–221.

Sudo, K., Sekine, S. and Grishman, R. (2001). Automatic pattern acquisition for japanese information extraction. In *Proceedings of the first international conference on Human language technology research*, (pp. 1–7). Morristown, NJ, USA: Association for Computational Linguistics.

Sudo, K., Sekine, S. and Grishman, R. (2003). An improved extraction pattern representation model for automatic ie pattern acquisition. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, (pp. 224–231). Morristown, NJ, USA: Association for Computational Linguistics.

Sugiyama, K., Tagawa, S. and Toda, M. (1981). Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man and Cybernetics*, *11*(2), 109–125.

Sundheim, B. M. (1992). Overview of the fourth message understanding evaluation and conference. In *Proceedings of the 4th Conference on Message Understanding*, (pp. 3–21). Morristown, NJ, USA: Association for Computational Linguistics.

Sundheim, B. M. (1995). Overview of results of the muc-6 evaluation. In *Proceedings of the 6th conference on Message understanding*, (pp. 13–31). Morristown, NJ, USA: Association for Computational Linguistics.

Surdeanu, M. and Harabagiu, S. M. (2002). Infrastructure for open-domain information extraction. In *Proceedings of the 2nd International Conference on Human Language Technology Research*, (pp. 325–330). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Swanson, D. R. (1987). Two medical literatures that are logically but not bibliographically connected. *Journal of the American Society for Information Science*, *38*(4), 228–233.

Swanson, D. R. (1991). Complementary structures in disjoint science literatures. In *Proceedings of the 14th Annual International ACM/SIGIR Conference*, (pp. 280–289).

Takeuchi, K. and Collier, N. (2002). Use of support vector machines in extended named entity recognition. In *Proceedings of the 6th Conference on Natural Language Learning*, (pp. 1–7). Morristown, NJ, USA: Association for Computational Linguistics.

TextAnalytics (2005). Text mining summit conference brochure. World Wide Web. URL `http://www.textminingnews/`

Twomey, S. and Horwitz, S. (2002). Chandra levy's remains found in park by dog. Washington Post. Last accessed 17/6/2008.
URL `http://www.washingtonpost.com/wp-dyn/content/article/2008/06/18/AR2008061801755.html`

Uchimoto, K., Nobata, C., Yamada, A., Sekine, S. and Isahara, H. (2003). Morphological analysis of a large spontaneous speech corpus in japanese. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, (pp. 479–488). Morristown, NJ, USA: Association for Computational Linguistics.

Vlastos, E. and Patel, A. (2008). An open source forensic tool to visualize digital evidence. *Computer Standards & Interfaces*, *30*(1-2), 8–19.

Walker II, J. (1990). A node-positioning algorithm for general trees. *Software - Practice & Experience*, *20*(7), 685–705.

Ware, C. (2004). *Information Visualization: Perception for Design*. Morgan Kaufmann.

Woodson, R. (2006). Starlight information visualisation technologies. Pacific Northwest National Laboratory.
URL `http://starlight.pnl.gov/overview.stm`

Yangarber, R., Grishman, R., Tapanainen, P. and Huttunen, S. (2000). Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the 18th conference on Computational Linguistics*, (pp. 940–946). Morristown, NJ, USA: Association for Computational Linguistics.

Young, P. (1996). Three dimensional information visualisation (survey). Computer Science Technical Report 12/96, Centre for Software Maintenance, Department of Computer Science, University of Durham.
URL `http://vrg.dur.ac.uk/misc/PeterYoung/pages/work/documents/lit-survey/IV-Survey/index.html`

Zaki, M. J. (2000). Scalable algorithms for association mining. *IEEE Transactions on Knowledge and Data Engineering*, *12*(3), 372–390.

Zhai, C., Velivelli, A. and Yu, B. (2004). A cross-collection mixture model for comparative text mining. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 743–748). New York, NY, USA: ACM.

Appendix A

Part of Speech Tag-Set

The text documents in this research are pre-processed using the Infogistics' NLProcessor POS tagger (Infogistics, 2001). NLProcessor determines the part of speech (POS) of each word in a sentence and applies a label or tag (e.g. noun, verb, adjective, etc.) for each word. A word may take several POS-tags and the correct tag is assigned based on its definition and context, e.g. the word "books" can function as a verb in the third person (VBZ) in the phrase "he books a reservation", or as a plural noun (NNS) in the phrase "he reads books".

A POS-tag is a morpho-syntactic feature belonging to one of two types of classes, based on the Penn Treebank Tag-Set (Marcus *et al.*, 1993). The tag-set divides words into open-class and closed-class words. The open-class categories contain the classes of words which are not limited to a finite number of words, such as nouns, adjectives, verbs and adverbs. The complete set of tags for the open-class categories are listed in Table A.1. The closed-class categories are listed in Table A.2 and consist of a finite and well-established number of words such as prepositions, articles, and wh-words (question words beginning with 'wh').

Table A.1: Modified Penn Treebank Tag-Set (open class categories).

POS Tag	Description	Example
JJ	adjective	green
JJR	djective, comparative	greener
JJS	adjective, superlative	greenest
RB	adverb	however, usually, naturally
RBR	adverb, comparative	better
RBS	adverb, superlative	best
NN	common noun	table
NNS	noun plural	tables
NNP	proper noun	John
NNPS	plural proper noun	Vikings
VB	verb base form	take
VBD	verb past	took
VBG	gerund	taking
VBN	past participle	taken
VBP	verb, present, non-3d	take
VBZ	verb present, 3d person	takes
FW	foreign word	d'hoevre

Table A.2: Modified Penn Treebank Tag-Set (closed class categories).

POS Tag	**Description**	**Example**
CD	cardinal number	1, third
CC	coordinating conjunction	and
DT	determiner	the
EX	existential there	there is
IN	preposition	in, of, like
LS	list marker	1)
MD	modal	could, will
PDT	predeterminer	both the boys
POS	possessive ending	friend's
PRP	personal pronoun	I, he, it
PRP$	possessive pronoun	my, his
RP	particle	give up
TO	to (both "to go" and "to him")	to go, to him
UH	interjection	uhhuhhuhh
WDT	wh-determiner	which
WP	wh-pronoun	who, what
WP$	possessive wh-pronoun	whose
WRB	wh-adverb	where, when

Appendix B

Named Entity Tags

In this dissertation named entity (NE) extraction was performed using the Java Extraction Toolkit (JET) (Grishman, 2007). JET detects and extracts selected types of entities mentioned in the source data. An entity is an object or set of objects in the world.

JET identifies the following seven types of entities:

- Person (PER) – Person entities are limited to humans. A person may be a single individual or a group.
- Organization (ORG) – Organization entities are limited to corporations, agencies, and other groups of people defined by an established organizational structure.
- Geo-political Entity (GPE) – Geo-political entities are geographical regions defined by political and/or social groups. A GPE entity subsumes and does not distinguish between a nation, its region, its government, or its people.
- Location (LOC) – Location entities are limited to geographical entities such as geographical areas and landmasses, bodies of water, and geological formations.
- Facility (FAC) – Facility entities are limited to buildings and other permanent man-made structures and real estate improvements.
- Vehicle (VEH) – A vehicle entity is a physical device primarily designed to move an object from one location to another, by (for example) carrying, pulling, or pushing

the transported object. Vehicle entities may or may not have their own power source.

- Weapon (WEA) – Weapon entities are limited to physical devices primarily used as instruments for physically harming or destroying other entities.

For each entity, JET records the type of the entity (PER, ORG, GPE, LOC, FAC, VEH, WEA), subtype, class, and all the textual mentions of that entity. The complete set of annotations for english and their explanations can be found in the *ACE (Automatic Content Extraction) English Annotation Guidelines for Entities* (LDC, 2005).

Appendix C

Force Based Layout Algorithm

The text graphs created in this research are displayed using a force-based algorithm, based on the algorithms of Eades (1984) and Churcher *et al.* (2004). The approach models pairwise forces of repulsion, F^r, between nodes and forces of attraction, F^a, between nodes connected by edges.

The repulsive force is calculated using Coulomb's law (equation C.1) which exhibits an inverse square law behaviour where the strength of the repulsive force between nodes i and j falls in proportion to the square of the distance between them. In practice, some slight modification is needed in order to prevent singularities where the distance r between nodes becomes very small. The coulomb constant, k^r, determines the strength of the repulsion. Thus, each node exerts a repulsive force on each other node, where the force depends only on the distance between the nodes. The magnitude of the force according to Coulomb's law is calculated by:

$$F_{ij}^r = \frac{k^r}{r_{ij}^2} \tag{C.1}$$

The attractive force is calculated using Hooke's law (equation C.2) which gives the attractive "restoring force" for elastic springs of natural length, l_0, when stretched by a spring constant, k^a, resulting in a displacement, x. The spring constant k^a represents the stiffness of the spring. The magnitude of the force according to Hooke's law is calculated by:

```
CoulombHookeLayout(Nodes, CoulombConstant, HookeConstant)
{
    foreach node node1 in Nodes
        netForce = (0, 0);

        foreach other node node2 in Nodes
            force = calcCoulombForce(node1, node2); //repulsion between nodes
            netForce += force;
        end loop

        foreach spring link in node1.Links //links are treated as springs
            force = calcHookeForce(node1, link); //attraction between connected nodes
            netForce += force;
        end loop

        //calculate the displacement of node1
        node1.position = node1.position + stepSize*netForce*damping
    end main loop
}
```

Algorithm C.1: Force based layout algorithm using Coulomb's law and Hooke's law.

$$K^a = -k^a(x - l_0) \tag{C.2}$$

The implementation of the algorithm described above is shown in pseudocode in Algorithm C.1.

Appendix D

Assignment of Weights to Units of Information

A master marking sheet was drawn up containing all of the relevant facts (characters, relations and events) extracted from the story. Weights were then assigned to each item based on an assessment of its importance to the story and the dificulty in extracting it. This assignment of weights to units of information is necessarily a subjective process as different persons may assign different levels of importance to elements of the plot or have differing opinions as to the dificulty involved in extracting certain information.

The manner in which these weights were assigned to the items of information from the text is set out in the tables below. Lists of the extracted locations and characters are presented in Table D.1 and Table D.2 respectively. A list of relations between characters and personal facts is set out in tables D.3 to D.4, followed by a list of extracted events in tables D.5 to D.9.

Table D.1: List of Locations.

Locations	**Weight**
Styles Court	1
Country place in Essex	1
In Styles St. Mary	1

Table D.2: List of Characters.

Characters	**Weight**
Mrs. Emily Inglethorp	5
Mr. Alfred Inglethorp	5
Mr. John Cavendish	5
Mrs. Mary Cavendish	5
Miss. Evelyn Howard	5
Hercule Poirot	5
Mr. Lawrence Cavendish	4
Mr. Hastings	4
Miss. Cynthia Murdoch	4
Dr. Bauerstein	4
Japp	3
Summerhaye	3
Dr. Wilkins	3
Annie	2
Mr. Wells	2
Mr. Philips	2
Sir. Ernest	2
Manning	2
William	2
Mrs Raikes	2
Dorcus	1
Elizabeth Wells	1

Table D.3: List of Relations and Personal Facts (a).

Relations and Personal Facts	**Weight**
Mrs Inglethorp has 2 step sons: John and Lawrence	2
Mrs. Inglethorp is married to Alfred Inglethorp	1
Alfred Inglethorp is a distant relation to Evelyn Howard	3
Scotland Yard Detectives: Japp and Summerhaye	2
Murderer had an accomplice	5
Dr Wilkins was Mrs Inglethorp's doctor	3
Housemaids at Styles Court: Dorcus, Annie, Elizabeth	1
Mr Wells was Mrs Inglethorp's attorney	2
Poirot is a belguim detective	2
Poirot is a friend of Hastings	1
Dr Bauerstein is an expert in poisons	2
Dr Bauerstein is a friend of Mary's	2
Cynthia has knowledge of poisons	2
Cynthia is a guest in the house	2
Cynthia sleeps in room next to Mrs Inglethorp	3
John and Lawrence both seem to like Cynthia	1
Cynthia thinks Mary hates her	1
Cynthia is the daughter of an old school fellow of Mrs Inglethorps	1
Cynthia is a protegee of Mrs Inglethorp's	2
Hasting is the narater	1
Hasting is a friend of John Cavendish	1
Hasting is an assistant to Poirot	2
Hastings has feelings for Mary Cavendish	1
Miss Howards hates Alfred	3
Miss Howard is Mrs Inglethorp's companion	1

Table D.4: List of Relations and Personal Facts (b).

Relations and Personal Facts	**Weight**
Lawrence is John's brother	2
Lawrence has no money	1
Lawrence qualified as a doctor	1
Mary works on the land	1
John practiced as a barrister	1
John settled at styles	1
John is married to Mary Cavendish	2
John is not a happy in his marriage	2
John received small allowance from his mother	1
John was struggling for money	2
John is the prisoner	4
Alfred Inglethorp is a distant relation to Evelyn Howard	3
Alfred has a distinctive look: black beard and leather boots	2
Alfred works as secretary for Emily Inglethorp	1
Alfred is young, 20 years younger than Mrs Inglethorp	1
Alfred is not liked by anyone, except Mrs Inglethorp	1
Alfred is thought to have married for money	1
Mrs Inglethorp lives at Styles Court	1
Mrs Inglethorp was a widow	1
Mrs Inglethorp was emotional, prone to fits of temper	1

Table D.5: List of Events (a).

Category	Event	Weight
Will	Gardeners witnessed new will	3
	Lawrence may inherit money from Mrs. Inglethorp	2
	John may inherit money from Mrs. Inglethorp	2
	Mrs Inglethorp wrote will in favour of Alfred	2
	Mrs Inglethorp cut her sons out of her will	1
	Mrs Inglethorp destroyed her new will	5
	Mrs Inglethorp intented to change her will	
	on the day of death	1
	Cynthia was not provided for in Mrs Inglethorp's will	2
Coffee/coco	Mrs I never drank her coffee - it was spilt	5
	coffee cup was smashed	2
	coco tested for strychnine, narcotic found	5
	stain and candle wax was found on	
	Mrs. Inglethorp's floor	1
	Annie takes coco to Emily	1
	Annie saw john take coffee to Mrs. Inglethorp	2
	Poirot noticed Cynthia's coffee cup was missing,	
	because she doesn't have sugar in coffee	4
	Mary hides Cynthia's coffee cup when she hears of poison	5
	Alfred poured Emily's coffee, but didn't give it to her	2
Poison	Poison was thought to be in the coffee	2
	Poison bought by "Alfred"	2
	strychnine sold my Mace	2
	narcotic delayed affect of poison	3
	bromine powder added to tonic which caused	
	the strychnine in it to precipitate	5
	Mrs Inglethorp was murdered by poison	2
	Poison identified as strychnine	2

Table D.6: List of Events (b).

Category	Event	Weight
Tonic/meds	Dr Wilkins perscribed tonic for Mrs. Inglethorp	3
	Cynthia sometimes makes sleeping draught for Mrs Inglethorp	1
Trial	Dr Wilkins testified at trial	2
	Dr Bauerstein testified at trial	1
	John was acquitted for the murder	5
	Poirot thinks John will be acquitted	1
Investigation	Poirot asked to quietly investigate	3
	Poirot solves the mystery	1
	Scotland Yard Detectives searched prisoner's room	2
	Dr Bauerstein is a suspect	2
	Dr Bauerstein arrested for espionage	4
	Miss Howard is suspected by Poirot of not telling the truth	2
	Miss Howard believes Alfred guilty	1
	letter to Evelyn Howard was the last link	5
	Miss Howard gave paper clue to Poirot	2
	Lawrence's finger prints were found on a bottle of strychnine in the dispensary	5
	Lawrence is suspected because he ordered a black beard	2
	John is suspected for the murder	3
	John is accused of pretending to be Alfred buying poison	5
	Investigators found poison in John's room	3
	John is arrested for the murder	2
	Alfred is the initial murder suspect	2
	Poirot didn't want Alfred arrested	4
	motive for murder: money from will	3

Table D.7: List of Events (c).

Category	Event	Weight
Surrounding the murder	Mrs Inglethorp	2
	Dr Bauerstein happened to be up on the night/morning of murder	3
	Dr Bauerstein came in the house for coffee on the night of murder	2
	Cynthia slept through noise of murder	3
	Cynthia was drugged by Mary so that Mary could sneak through her room to Mrs Inglethorp's room	8
	Miss Howard disguised herself as Alfred	10
	Miss Howard hoped to pin the murder on John	5
	Miss Howard planned murder/bromides	4
	Miss Howard was on duty night of murder	2
	Miss Howards had an argument with Mrs. Inglethorp	1
	Miss Howards received letter from Mrs. Inglethorp on 7th/17th	3
	Mary drugs Mrs. Inglethorp's coco to make her sleep	5
	Mary was in Mrs Inglethorp's room night of murder	5
	Mary's green land armlet was caught in the door bolt of Mrs Inglethorp's room	3
	Mary spilt candle max on Mrs Inglethorp's floor	3
	Mary was looking for paper proving her husband's infedility in Mrs Inglethorp's room	5
	Mary thinks Mrs Inglethorp has proof of her husband's infedility	4
	Mary lied about hearing table fall	3
	Mary overhears her husband's quarrel with Mrs Inglethorp	4

Table D.8: List of Events (d).

Category	Event	Weight
Surrounding the murder	Mary thought she had mistakenly killed Mrs Inglethorp	3
	Mary quarrelled with Mrs Inglethorp	1
	John quarrelled with Mrs Inglethorp the day before she was murdered	4
	Alfred murdered his wife, Mrs Inglethorp	20
	Alfred was out walking - Monday 16th	5
	Alfred was accused of quarrelling with Mrs Inglethorp on Tuesday 17th	2
	Alfred said wife mistook Bauersyein for him when she said her last words	2
	Alfred was out night of murder	2
	Mrs Inglethorp died on tues night	2
	Night of murder was the 17 July	2
	Mrs Inglethorp was meant to be murdered on the Monday night	3
	The night before the murder, Mrs Inglethorp's bell wire was cut	2
	Cynthia was meant to be out at time of murder	2
	The doors were locked from inside of Mrs Inglethorp's room	1
	Mrs Inglethorps last words were "Alfred Alfred"	1
	Mrs Inglethorp was agitated - she found a letter planning her death	5
	Mrs Inglethorp found a letter when looking for stamps, the letter upset her	2
	Mrs Inglethorp forgot to take her medicine/tonic on Monday night	3

Table D.9: List of Events (e).

Category	Event	Weight
Other	Cynthia works in the dispensary at the Red Cross Hospital in Tadminster	2
	Hastings stayed at Styles often in his youth	1
	Hastings was wounded from the war, and was recovering at styles	1
	Hastings was jealous of Dr. Bauerstein	1
	Hastings offers to mary Cynthia	1
	Hastings is visiting at styles	1
	Mary spends lots of time with Dr. Bauerstein	2
	Mary was trying to make husband jealous	3
	Mary appears to be in love with Bauerstein and is thought to be having affair	1
	John had an affair with Mrs Raikes	3
	John asked Mary not to see Bauerstein	1
	Alfred intended to elope with Evelyn Howard	10
	rumour of affair between Alfred and Mrs. Raikes	1

Appendix E

Acronyms

The acronyms used throughout this dissertation are listed below in alphabetical order.

GLARF	Grammatical and Logical Argument Framework
GUI	graphical user interface
idf	inverse document frequeny
IE	information extraction
IR	information retrieval
JET	Java Extraction Toolkit
KDD	knowledge discovery in databases
LSI	latent semantic indexing
MUC	Message Understanding Conference
NE	named entity
NER	named entity recognition
NLP	natural language processing
POS	part of speech

TDT Topic Detection and Tracking

tf-idf term frequency - inverse document frequeny

Appendix F

Symbols

The symbols used throughout this dissertation are defined below, grouped under the chapter in which they first appear.

F.1 Chapter 3: Survey of Text Mining Methods

c	The confidence of a rule
χ^2	Chi Squared
D	Represents the set of all documents
d_i	The i^{th} document
df	The document frequency
$\emptyset$	The empty set
F	The F-score
f	Frequency
I	The set of all items
i_i	The i^{th} item
idf	The inverse document frequency
$\cap$	Intersection
MI	Mutual information
N	The size of the dictionary

P	Probability
Φ^2	Phi Squared
$\Rightarrow$	Rule in the form of an implication
s	The support of a rule
$\subseteq$	A subset of or equal to
T	Represents the set of all the words in the dictionary
t_i	The i^{th} word, or term
tf	The term frequency
$\cup$	Union

F.2 Chapter 5: Relation Discovery Models

n_k	The number of links for a node
T_{assoc}	The association ratio threshold
T_{freq}	The frequency threshold

F.3 Appendix C: Force Based Layout Algorithm

F^a	The force of attraction by edges
F^r	The force of repulsion by nodes
k^a	Spring constant
k^r	Coulomb constant
l_0	The natural length
r	The distance between nodes
x	The displacement resulting from a force

Index