Hypothesis generation and testing in event profiling for digital forensic investigations

DOI: 10.4018/jdcf.2012100101

Hypothesis Generation and Testing in Event Profiling for Digital Forensic Investigations

Lynn Batten, Faculty of Science and Technology, School of IT, Deakin University, Burwood, Melbourne, VIC, Australia

Lei Pan, Faculty of Science and Technology, School of IT, Deakin University, Burwood, Melbourne, VIC, Australia

Nisar Khan, Faculty of Science and Technology, School of IT, Deakin University, Burwood, Melbourne, VIC, Australia

ABSTRACT

The need for an automated approach to forensic digital investigation has been recognized for some years, and several authors have developed frameworks in this direction. The aim of this paper is to assist the forensic investigator with the generation and testing of hypotheses in the analysis phase. In doing so, the authors present a new architecture which facilitates the move to automation of the investigative process; this new architecture draws together several important components of the literature on question and answer methodologies including the concept of 'pivot' word and sentence ranking. Their architecture is supported by a detailed case study demonstrating its practicality.

Keywords: *Digital Forensics, Event Profiling, Evidence Extraction, Hypothesis Generation, Ranking*

INTRODUCTION

In practice, digital forensics is carried out with the aim of extracting evidence which will be tenable in a court of law (Carrier, 2006; Willassen, 2008). A stream of research work in the last decade has attempted to assist the forensic investigator in moving from the historically manual approach towards an automated, and therefore also reproducible, approach to the discovery of digital evidence (Batten & Pan, 2011; Jankun-Kelly, Wilson, Stamps, Franck, Carver, & Swan, 2009; Marrington, Mohay, Morarji, & Clark, 2010; Pan, Khan, & Batten, 2012). Carrier (2006) and Marrington (2009) both developed automated methods of describing a computer system and its activity over a fixed period of time; the former focused on the raw data while the latter focused on events surrounding a crime. Both authors look for relationships between the objects they are examining. The work of Batten and Pan (2011) and Pan, Khan, and Batten (2012) extends the work of both Carrier (2006) and Marrington (2009) by

DOI: 10.4018/jdcf.2012100101

demonstrating how relationships between the objects of investigation can be used to reduce the size of the data set needing analysis and so speed up the investigation time.

All of Batten and Pan (2011), Carrier (2006), Marrington (2009), and Pan, Khan, and Batten (2012) develop extensive methodologies for relationship building. Carrier (2006) gives examples of hypotheses which can be formulated and tested; however, he does not attempt to define the word hypothesis. The authors of Al-Zaidy, Fung, Youssef, and Fortin (2012) use a similar method of relationship building and develop 'hypotheses' in the form of relationships between people and data; however, again, the authors do not define formally what they mean by a hypothesis.

An important contribution of Pan, Khan, and Batten (2012) is a formal definition of hypothesis in the context of digital forensic investigation and an illustration of how the theoretical formulation is able to find relationships from which hypotheses can be developed and examined. In this paper, we move to a new level in investigating the relevance of hypotheses to the situations at hand. We continue to automate the analysis as much as possible in order to apply rigor to the methodology and to provide the ability to replicate the methodology as needed for the court.

First, we describe the relevant literature. The section afterwards contains formal definitions and notations needed to illustrate our subsequent work and we discuss the hypothesis generation and testing methods in detail. A case study is presented and analyzed next; this case study is a continuation of that used in Batten and Pan (2011) and Pan, Khan, and Batten (2012). Finally, we summarize the implications of our work and consider its impact on the future research literature in this area.

RELATED WORK

The paper by Radev, Prager, and Samn (2000) deals with answering natural language questions. In this paper the authors introduce a method called 'predictive annotation' which highlights phrases in the text in advance by assigning labels to make these phrases the targets of a particular question. When dealing with Natural Language Processing, there are many questions that contain words which, when searched for in a corpus may not be returned or answered as they may not exist in the corpus. The concept of 'predictive annotation' was introduced in Radev, Prager, and Samn (2000) precisely to deal with such situations. This method works in two steps. In the first step, the question under consideration is enhanced by assigning labels (which the authors call QA-Tokens) to a set of 'recognized' objects such as places or persons; then, the text in the corpus is labelled with the same QA- Tokens for all recognized objects. Finally, the QA-Tokens in the question are searched for in the corpus to locate matching passages. The system architecture for this step has two components: the Information Retrieval component returns a list of 10 short passages containing a large number of potential answers for each query, and the Answer Selection component which ranks the potential answers using the two algorithms AnSel and Werlec. Both algorithms will return five text passages per query that contain the possible answers. This means that, in the initial stage, no single answer will be returned but rather a list of possible or potential answers.

In the second step, the answer selection process inputs the matching passages identified in the first step and ranks them. In ranking, a weighting scheme is adopted to assign weights to the text. A weight of 400 is assigned to QA-tokens; proper nouns get a weight of 200; and any other words get a weight of 100. In addition, a score from 1 to 99 is assigned to text passages based on how close the matching query tokens are to each other within a passage based on a given definition of proximity.

In the paper Prager, Duboue, and Chu-Carroll (2006), the idea of question inversion is introduced to improve the ranking accuracy. Inversion is applied to questions which are generally about relations between objects or entities. It has been argued that, when searched for in a corpus, an inverted question will return a different set of passages as compared to the passages returned by the original question. The authors use the concept of inversion to increase or reduce support for a candidate answer. To invert a question one must identify an object or entity in the question of a known type; this entity is referred to as a 'pivot term' in Prager, Duboue, and Chu-Carroll (2006). However in this work the questions are not inverted in the form of natural language but rather are presented in the context of a new concept introduced by the authors and referred to as a QFrame. A QFrame comprises three entities: Keywords (the list of terms and phrases in the question which have known 'type' or 'entity type' e.g., time, country, etc.), AnswerType (the entity type for the answer sought in the question), and relationships (the set of relationships among the Keywords and the AnswerType). For instance, based on type 'name' (or 'who'), in the question 'What is the capital of Australia?' The word Australia would be identified and the question then inverted to 'Canberra is the capital of what country?'

The focus of the paper by Kalyanpur, Patwardhan, Boguraev, Lally, and Chu-Carroll (2012) is on the decomposition of fact-based questions. The authors suggest that decomposing fact-based questions will produce more reliable and correct final answers as these will be supported by other facts. In the paper, questions can be decomposed in either a parallel or nested way. The authors argue that questions which are decomposable in parallel hold completely independent facts, while nested decomposable questions hold dependent facts about an entity which, when combined, link the entity to the correct answer. The decomposition framework is composed of four key components: decomposition recognizers, question rewriters, an underlying QA system, and candidate rerankers. The system was evaluated on nearly 3000 sets of complex Final Jeopardy questions from which it was determined that the results of parallelly decomposed questions were more significant than those for nested decomposable questions (likely due to the fact of dependence on sequencing in nested questions).

According to Chu-Carroll, Fan, Schlaefer, and Zadrozny (2012) the delivery of a question-answering (QA) system is highly dependent on the quality of available resources to be analyzed by the system for answering a question. The authors explain that most QA systems operate on a task-oriented corpus and hence, are not qualified for handling questions such as in the *Jeopardy!^{TM}* program. To deal with this gap, three procedures are developed to produce a self-sufficient, reliable, and relevant text corpus. These procedures are source acquisition (the process of obtaining new documents), source transformation (the process of transforming the extracted information from the available resources into a form which the system can easily handle), and source expansion (the process of augmenting the scope of each known topic by incorporating new information to it). The results show a significant shift in accuracy (from 59% to 70.4%) of correct answers.

In Murdock, Fan, Lally, Shima, and Boguraev (2012) an innovative technique called Supporting Evidence Retrieval is introduced. This technique identifies the relationship between the candidate answer (which the authors call candidate hypothesis) and the question. This is done by inserting candidate answer terms back into the question to form a new statement which is then searched again. It was shown that by executing further searching, this method delivers more passages that contain candidate answers which are related to the question but in a different manner. This helps in scoring and ranking to choose for the correct answer. The experimental results demonstrate a 6.9% improvement over their baseline target.

Modern QA systems deeply interact with natural language questions. According to Chu-Carroll, Brown, Lally, and Murdock (2012), answering natural language questions might be deceiving as these might contain concealed associations and inferred or tacit correlations or relationships. The authors adopted a 'spreading-activation' approach to identify the unknown relationships behind a question. The focus is on questions that inquire about concealed relationships between multiple entities and missing-links. The knowledge resources used for experimentation were n-gram theory, prismatic knowledge and Wikipedia links. Accuracy and precision using this method were improved by10% and 11% respectively.

In QA systems much attention has been given to discerning semantic relations in the text corpora as these relations can be used to highlight possible answers and also act as support for text passages. To discern the relationships in this regard, the authors of Wang, Kalyanpur, Fan, Boguraev, and Gondek (2012) use two approaches: rule-based relation extraction and statistics-based relation extraction.

The hypothesis generation phase is critical to the architecture of IBM's Watson. According to Chu-Carroll, Fan, Boguraev, Carmel, Sheinwald, and Welty (2012), hypotheses are the promising or plausible answers to the questions. Hypothesis generation is carried out with the aid of two components: a search component delivers related items to the question from the unstructured and structured knowledge sources followed by which, candidate generation highlights promising answers to the questions.

In the next section, we introduce our concept of hypothesis generation formally in a well-structured manner for digital forensic investigation. From our point of view, a hypothesis is an English language statement, and so the QA approaches described above are applicable in our context.

HYPOTHESIS GENERATION AND TESTING

We refer the reader to the paper by Pan, Khan, and Batten (2012) for the discussion on our formal approach to hypothesis generation which is based on the work in Batten and Pan (2011) and Marrington (2009).

We recall the notation here. We begin with a set of objects **O** which have been collected in the preliminary stage of the investigation; relationships are then established between some of these objects. For instance, given objects *Alice* and *printer*, we might establish the relationship that '*Alice printed something on the printer.*'

Notation and Terminology

As in Pan, Khan, and Batten (2012), **O** is the set of items perceived to be in the vicinity of, or connected to, a forensic investigation. We again use the standard definitions 1 through 4 below which can be found in Herstein (1975) or our paper Pan, Khan, and Batten (2012).

Definition 1: *A relation* $\boldsymbol{R}$ *on* $\mathbf{O}$ *is a subset of ordered pairs of* $\mathbf{O}\times\mathbf{O}$.

Definition 2: *A relation* $\boldsymbol{R}$ *on* $\mathbf{O}$ *is reflexive if* $(a,a) \in \boldsymbol{R}$ *for all* a *in* $\mathbf{O}$.

Definition 3: *A relation* $\boldsymbol{R}$ *on* $\mathbf{O}$ *is symmetric if* $(a,b) \in \boldsymbol{R}$ *implies* $(b,a) \in \boldsymbol{R}$ *for all objects* a *and* b *in* $\mathbf{O}$.

In our context we can assume that any relation on **O** is both symmetric and reflexive since these properties have no effect on information in a forensic investigative sense.

Definition 4: *Given a reflexive and symmetric relation* $\boldsymbol{R}$ *on* $\mathbf{O}$, *for each element* $a \in \mathbf{O}$, *we define a relational class for* a *by* $(a) = \{b \mid (a,b) \in \boldsymbol{R}, b \in \mathbf{O}\}$.

Note that, because of reflexivity, $a \in \mathbf{O}$ is always an element of the relational class (a). Also, if $b \in (a)$, then $a \in (b)$ by symmetry. In addition, we adopt the definition of hypothesis introduced in Pan, Khan, and Batten (2012):

Definition 5: *A* ***hypothesis*** *h about* $\mathbf{O}$ *and* $\boldsymbol{R}$ *is a statement involving a non-empty subset* $\mathbf{O}_h$ *of* $\mathbf{O}$ *such that for all* $a \in \mathbf{O}_h$, *if* $|\mathbf{O}_h|>1$, *then there is an element* $b \in \mathbf{O}_h$ *with* $b \neq a$ *such that* $(a,b) \in \boldsymbol{R}$.

Hypotheses are therefore generated using objects from a non-empty subset of the object set; if there is more than one object present, then it must be related to some different object also used in the hypothesis. That hypotheses should be about objects which are related is an important constraint.

Example: Suppose that {Alice, printer, document} is a subset of **O** and that **R** contains (Alice, printer) and (printer, document). Then the statement: 'Alice printed the document on the printer' satisfies definition 5. The statement 'Alice did not print the document on the printer' is also a valid hypothesis; so it is only within a larger context that we will be able to say which is more likely to be correct.

Object Types

Given an object set, we identify the type of some of the words in it in order to generate hypotheses for analysis in the hypothesis testing phase. In this regard, we follow the authors of Prager, Duboue, and Chu-Carroll (2006) who use 'pivot terms' to generate what they call an 'inverted question' from a given one, in so doing, producing redundancies which can improve a system's ability to find a correct answer.

A word type will depend on the object set used and will be generic as for example a word which can be classified as 'who' or 'what.' As an example consider the object set {Alice, John, File, printer 1, printer 2} with word types 'name' and 'device' Then in the hypothesis 'Alice printed the document on printer 1,' the words 'Alice' and 'printer' correspond to identified word types. The pivot methodology would consider all hypotheses generated by replacing 'Alice' by 'John' and 'printer 1' by 'printer 2,' thus generating a total of four hypotheses.

For simplicity, we can reduce word types to a standard set of five: 'who,' 'what,' 'where,' 'when,' and 'how.' In this case, a name falls under 'who' and a device under 'what.' The 'how' type needs a more careful formulation as it is not quite as precise as the other four types.

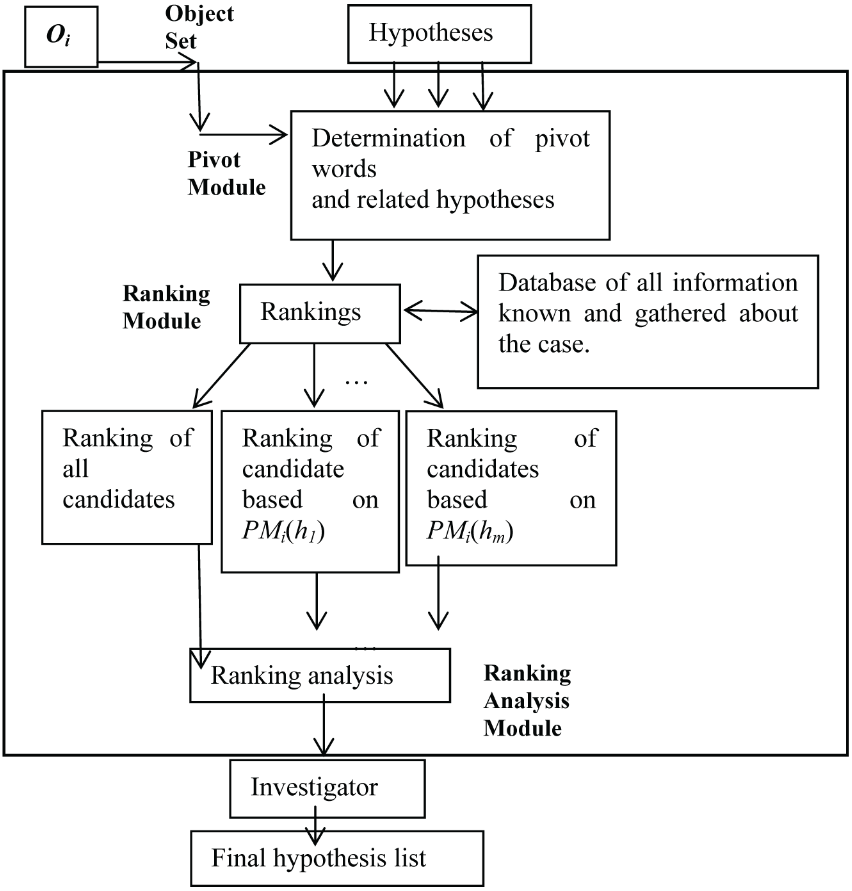
In some schemes, such as Harabagiu, Hickl, and Lacatusu (2006), the addition of opposite or negative statements to the set to be ranked is viewed as being helpful. However, in our case, a very low ranking on a sentence probably means a high ranking on its negative, and conversely. We therefore argue that it is not worthwhile to use both. For instance, a low ranking on 'Joe used the 90 JPG files' probably indicates that 'Joe did not use the 90 JPG files.' So we get this second sentence ranked for free as it were. Based on this argument, we do not automatically include negative statements of those included into the set to be ranked.

Analyzing the Hypotheses

In our paper Pan, Khan, and Batten (2012), once hypotheses are generated, the investigator analyzes and assigns confidence levels to them. In the current paper, we automate this phase to a large extent. Our work is influenced by the fact that extremely good ranking systems for question-answer type problems already exist. Three such examples are Ferrucci (2012), Gondek et al. (2012), and Li (2011). We also use the 'pivot' concept of Prager, Duboue, and Chu-Carroll (2006) in dealing with increasing the accuracy of question-answer systems; the idea here, is to identity the 'type' of certain words in a sentence and to invert the sentence around each such word. (We gave an example earlier in this section.)

In our situation, hypotheses are derived based on an object set and we identify some of the words in the object set by type, for instance 'who' and 'what' as described earlier in this section. Then possibly redundant hypotheses are generated from the Pivot Module (Figure 1). Those hypotheses are then all ranked using a ranking module which has access to a database of

Figure 1. Architecture of our hypothesis ranking scheme


O_i
Object Set
Hypotheses
Pivot Module
Determination of pivot words and related hypotheses
Ranking Module
Rankings
Database of all information known and gathered about the case.
…
Ranking of all candidates
Ranking of candidate based on $PM_i(h_1)$
Ranking of candidates based on $PM_i(h_m)$
…
Ranking analysis
Ranking Analysis Module
Investigator
Final hypothesis list

information, and the top-ranked hypotheses are considered by the investigator. We describe the components of our architecture in more detail.

Pivot Module

An investigator will assess the case using several rounds, as described in detail in our case study in the "Case Study and Analysis" section. After several rounds, a number of hypotheses $h_{1,} h_2 \cdots h_m,\ m \geq 1$ will have been generated and the investigator now pauses to evaluate them using the scheme described in Figure 1.

The Pivot Module (PM) inputs each hypothesis or sentence generated at this stage along with the current object set O_i in use, identifies the word types in the sentences and outputs an enlarged set of sentences.

Notation

If *h* is a specific hypothesis input to the Pivot Module based on an object set O_i, then $PM_i(h)$ is the set of sentences resulting from the application of the *PM* on *h*.

Returning to the example in the sub-section on Object Types, PM_i (Alice printed the document on printer 1) is a set of four sentences. We have been careful to use the word 'sentence' here rather than 'hypothesis' since these new sentences may fail to meet the part of our technical definition which requires that for all $a \in O_i$, if $| O_i |> 1$, then there is an element $b \in O_i$ with $b \neq a$ such that $(a, b) \in R$.

Ranking Module

The Ranking Module (*RM*) now takes as input all sentences produced by the *PM*. For the *RM*, receiving sentences rather than hypotheses is not a problem as we assume that it has access to a comprehensive database in which all information possibly related to the case has been collected. Such a database may, for instance, include evidence of a witness who claims to have seen Alice print, while no evidence is present to indicate that John printed. This should raise the level of confidence with which Alice can be asserted to have done the printing.

The RM in our case study is an off-the-shelf product (Ferrucci, 2012; Li, 2011) known to provide reliable results. On input of a set of $m \geq 1$ sentences, the RM ranks them from 1 (most likely) to *m* (least likely) based on their likelihood of being correct given the database of information. For each h_i, $i = 1$, *m*, *RM* outputs a ranking of the set $PM_j(h_i)$ for the corresponding object set O_j, as well as a ranking of $\bigcup_{i=1}^{m} PM_j(h_i)$. (Note that this union may contain repeated sentences.)

Ranking Analysis Module

The Ranking Analysis Module (RAM) is another off-the-shelf product which ranks natural language sentences. One such product, described in Radev, Prager, and Samn (2000) ranks possible answers to a fixed question. The scheme is easily adapted to our use as we can rewrite sentences as questions. For example, the sentence 'Alice printed the document on the printer' can be reworded as 'did Alice print the document on the printer.'

The final rankings are now output to the investigator who may first look at the overall ranking and then examine the rankings of $PM_j\left(h_1\right)$, $PM_j\left(h_2\right)$ and $PM_j\left(h_3\right)$, where h_1, h_2 and h_3 are the top three overall, to see if there are any inconsistencies with these rankings. She then decides if the case is close to being resolved or if more rounds need to be executed.

As an example, Table 1 shows output of the RAM which would be considered to be contradictory. Two initial hypotheses h_1 and h_2 were input to the scheme, each generating four sentences $h_1^1 = h_1$, h_1^2, h_1^3 and h_1^4, $h_2^1 = h_2$, h_2^2, h_2^3 and h_2^4.

While h_1^3 and h_2^4 are ranked top overall, these are both in the lower half of their rankings in their Pivot Module sets. An investigator would therefore be uncomfortable in proposing h_1^3 or h_2^4 as solutions. On the other hand, a particular sentence might be highly ranked both in its own pivot module set and across the entire set of candidates; in this case, it might well be proposed as a solution.

In the next section, we consider an extension to the case study used in Pan, Khan, and Batten (2012) to incorporate the architecture of Figure 1. Based on time and resources, the investigator determines a number of rounds to be run, and, in each round, generates a set of hypotheses. After several hypotheses have been collected, these are ranked and the investigator may choose to keep only those with a high ranking. If the case cannot be wrapped up immediately based on these high-ranking statements, then more rounds can be run.

CASE STUDY AND ANALYSIS

Case studies have often been found to be an exemplary way of demonstrating the validity of a theory, as was advocated in Batten and Pan (2008). In order to illustrate our new architecture, we continue with the case study used in Batten and Pan (2011) and Pan, Khan, and Batten (2012). In our previous case study in Pan, Khan, and Batten (2012) and Batten and Pan (2011), a drug trafficker, Joe, hid his customers' contacts in steganographic picture files. Police

Table 1. Output from the RAM

Overall	$PM(h_1)$	$PM(h_2)$
h_1^3	h_1^1	h_2^1
h_2^4	h_1^2	h_2^3
h_1^2	h_1^3	h_2^2
h_2^3	h_1^4	h_2^4
h_1^4		
h_2^2		
h_1^1		
h_2^1		

officer Moti analyzed a forensic copy of Joe's laptop; Moti collected a set of file objects and built various relations based on the object sets. Furthermore, Moti formulated hypotheses based on these objects and their relations. After two days' endeavour, Moti managed to recover 20 steganographic pictures, two of which were encrypted with Joe's medicare card number. The encrypted contents contained the name of one of Joe's customers along with his mobile number. Whilst Joe was held in custody, the police carried out an in-depth investigation in the hope of locating the drugs trafficked by Joe; Joe consistently claimed his innocence. The scenario below continues from the previous one; the investigative team has identified Adam, a man who appears to do deliveries for Joe.

Adam drives a minivan registered in Joe's name. After Adam lost contact with Joe in late September 2009, Adam became nervous and deleted many files and records related to Joe's unlawful business. The following events happened chronologically:

2009-10-01 03:30 Adam switched on his laptop in his bedroom.
2009-10-01 03:31 Adam successfully con nected to the Internet.
2009-10-01 03:32 Adam launched his email program Windows Mail.
2009-10-01 03:40 Adam selected all the emails between him and Joe.
2009-10-01 03:50 Adam deleted the selected emails in Windows Mail to the Windows recycle bin.
2009-10-01 03:51 Adam launched the pro gram WinRAR and selected all documents in a folder named "Joe's work."
2009-10-01 04:00 Adam used WinRAR to compress the selected documents to a RAR file named "joe.rar" on the Desktop, and then he encrypted this file with a password.
2009-05-01 04:03 Adam concatenated the RAR file to a JPG file named "1.jpg."
2009-10-01 04:04 Adam moved all the documents in "Joe's work" to the recycle bin.

2009-10-01 04:05 Adam plugged a USB thumb drive into his laptop and moved the file “1.jpg” to the USB drive.
2009-10-01 04:07 Adam ran the program Eraser to overwrite the files in the recycle bin with randomly generated data.
2009-10-01 04:11 Adam shut down the laptop so that all cached data in the RAM were removed.

A few days later, the police seized Adam’s laptop and other belongings under a search warrant issued by the judge of Joe’s drug trafficking case. As part of a formal investigation procedure, police officers made a forensic image of the hard disk of Adam’s laptop and transported this laptop along with other devices found near it into the evidence locker. But Adam refused to answer any questions about Joe’s case. Because of the close relationship between Joe and Adam, the forensic team decides to assign Moti to investigate Adam’s electronic devices.

As a routine practice in beginning an investigation, Moti runs Forensic ToolKit to filter out the files of known hash values from a verified forensic image of Adam’s laptop. The filtered results are 1932 emails, 3060 JPG files and 8 application programs. Then he defines $\mathbf{O}$= {1932 emails, 3060 JPG files, 8 application programs} as his initial object set. To avoid analyzing all data bit by bit, Moti adopts our hypothesis generation approach which works in multiple rounds.

Round 1: Suspecting that Adam communicates with Joe via email, Moti writes his first hypothesis as h_1=“There are emails from Joe among the 1932 emails.” Since this new case aims to seize Joe’s drugs, Moti adds the identified suspect’s name “Joe” to establish a new object set as $\mathbf{O_1}$= {1932 emails, Joe}. Moti identifies the pivot words in h_1 as “Joe” and “1932 emails” both of which are in the object set. Then Moti associates types to the pivot words – “Joe” has type “who” and “1932 emails” has type “what.” Moti sets relational class $\mathbf{R_1}$= {(a,b) | a∈{1932 emails}, b∈{Joe}}. According to this setup, Moti supplies Joe’s name and email addresses as search criteria to perform a keyword search through the 1932 emails, but he finds no positive results.

Now Moti uses our ranking scheme to continue the investigation. By substituting for the identified pivot words in h_1, Moti derives the following candidate sentences: $h_1^1=h_1$, h_1^2= “Some of the emails from Joe are in the 1932 emails but there are others somewhere else,” h_1^3= “The emails from Joe are hidden in the 3060 JPG files,” and h_1^4= “There are no emails from Joe in the 1932 emails but there are emails from Joe somewhere else.” Adam’s denials of his connection to Joe suggest that he has enough motivation to destroy some key evidence; and Adam had a few days before being detained to permit him to modify the items on his laptop. Moti obtains the following ranked results $h_1^4>h_1^3>h_1^2>h_1^1$ from the pivot module output $PM_1(h_1)$ = {h_1^4= “There are no emails from Joe in the 1932 emails but there are emails from Joe somewhere else,” h_1^3= “The emails from Joe are hidden in the 3060 JPG files,” h_1^2= “Some of the emails from Joe are in the 1932 emails but there are others somewhere else,” h_1^1= “There are emails from Joe among the 1932 emails”}.

Round 2: Knowing the facts that emails are kept on both sender’s and recipient’s computers, and that Joe’s laptop is in custody, Moti writes his second hypothesis as h_2 = “The emails to Adam from Joe are in Joe’s laptop.” Moti establishes his object set as $\mathbf{O_2}$= {Joe’s emails, Joe’s laptop}. Moti identifies the pivot words as “emails” and “laptop.” In particular, “emails” has type “what,” and “in Joe’s laptop” has type “where.” And the relational class becomes $\mathbf{R_2}$= {(a,b) | a∈{Joe’s emails}, b ∈{Joe’s laptop}}. Moti then signs an evidence form and acquires a forensic image of Joe’s laptop. Having performed a similar search as the one in Round 1, Moti recovers 20 emails between Joe and Adam from Joe’s laptop. The fact that these emails are not present on Adams’ laptop triggers Moti’s suspicion that Adam erased the emails and that they are relevant to this case. So, Moti retains the hypothesis h_2.

Round 3: Suspecting that Adam erased the emails from Joe from his laptop, Moti generates a new hypothesis h_3= "Adam erased the 20 emails by using a program on his laptop." Moti includes the 20 emails found on Joe's laptop and the 8 programs on Adams' laptop in the object set $\mathbf{O_3}$= {20 emails, 8 application programs}. He identifies the pivot words as "emails" and "programs." The phrase "the 20 emails" has type "what," and "by using a program" has type "how." Moti sets the relational class $\mathbf{R_3}$= {(a,b) | a ∈{20 emails} and b∈{8 application programs}}. On Adam's laptop, Moti discovers that two programs can be used to delete email messages – Windows Mail and Eraser. So Moti substitutes these two programs from $\mathbf{O_3}$ into the hypothesis h_3 to obtain two new hypotheses h_3^1= "Adam erased the 20 emails by using Windows Mail," and h_3^2= "Adam erased the 20 emails by using Eraser." Then our ranking module ranks h_3^1 higher than h_3^2 because Windows Mail can be used to delete individual email messages with a single click but the Eraser program cannot be easily used to delete individual emails from an inbox file.

This is almost the end of the first working day, and Moti uses our ranking scheme to summarize the first day's work. So he enters his hypotheses into the ranking analysis module and obtains the results shown in Table 2.

The Overall column provides the overall ranking of all Moti's hypotheses in the first three rounds. Specifically, the hypotheses are sorted according to the likelihood of being true - in this case, the confirmed hypothesis h_2 is ranked on the top; the unconfirmed ones h_3^1 h_3^2, h_1^4 and h_1^2 are in the middle; and the bottom two hypotheses h_1^3 and h_1^1 are most likely to be untrue. Note that h_1^2 is ranked higher than h_1^3 in the overall ranking, probably due to the fact that Moti has found no relevant information in the 20 recovered emails between Joe and Adam.

The next three columns $PM_1(h_1)$, $PM_2(h_2)$ and $PM_3(h_3)$ provide individual rankings over individual hypotheses. These columns preserve the orders of the ones derived in each round.

Since the current findings lead to no further conclusions, Moti decides to extend the investigation for three extra rounds in the second working day and decides to work on the top three hypotheses h_2, h_3^1, and h_3^2 during the next stage. Since Moti continues without any first round hypothesis, he ignores $PM_1(h_1)$ during the next three rounds.

Table 2. Output from the RAM after Round 3

Overall	**$PM_1(h_1)$**	**$PM_2(h_2)$**	**$PM_3(h_3)$**
h_2	h_1^4	h_2	h_3^1
h_3^1	h_1^3		h_3^2
h_3^2	h_1^2		
h_1^4	h_1^1		
h_1^2			
h_1^3			
h_1^1			

Round 4: In order to prove that Adam used the program Eraser to wipe off evidence from the laptop's hard drive, Moti decides to restore the timeline of the destruction of evidence. The success of reconstructing a correct timeline for digital crimes depends on information in trustworthy log entries which Moti can extract from Adam's laptop. Moti then writes his fourth hypothesis as h_4= "Adam deleted the emails from Joe in late September or early October 2009." Knowing that deleting a file generally does not affect system events, Moti sets his object set $\mathbf{O}_4$= {Windows events on Adam's laptop, September to October 2009}. The pivot words are "events" and "September to October 2009" of type "what" and "when" respectively. Moti sets the relational class $\mathbf{R}_4$= {(a,b) | a ∈{Windows events on Adam's laptop}, b ∈{September to October 2009}}. Then he uses the program LogParser to extract time information from the Windows event files on Adam's laptop. Using the system startup and shutdown events (EventID= 4608 or 4609) as filters, Moti notices an unusual pair of events which took place early in the morning of October 1, 2009. From the search results of LogParser, it was evident that Adam mostly used his laptop during daytime or early evening; however, Moti finds start-up and shut-down events at 03:30 am and at 04:11 am on October 1, 2009. Furthermore, Moti recalls that installing the Eraser program leaves an "InstallDate" subkey entry in the Windows Registry where the MSI installer records the installation time of Eraser. By examining this Registry subkey, Moti confirms that the early morning activity is suspicious when he finds that the program Eraser was installed on September 29, 2009 on Adam's laptop. Moreover, Moti is able to determine that a USB device was plugged into the laptop at 04:05 am on October 1, 2009. Moti becomes increasingly confident that the above dates and times are important; but he still needs further evidence. So, he retains his hypothesis h_4= "Adam deleted the emails from Joe in late September or early October 2009."

Round 5: Based on the forensic findings in the previous four rounds, Moti is confident that Adam wiped off the files related to Joe prior to his arrest. However, since these files could be too valuable for Adam to lose, Moti hypothesizes that Adam may have backed them up to external storage media before erasing them from the laptop. So he writes h_5= "Adam moved the files related to Joe to external storage media." Knowing that plugging and unplugging external storage devices leave a footprint on Windows Registry, Moti includes the Registry into his object set $\mathbf{O}_5$= {the files related to Joe, external storage media devices, Windows Registry on Adam's laptop}. The phrase "the files related to Joe" has type "what" and "to external storage media devices" has type "where." Moti sets the relational class $\mathbf{R}_5$= {(a,b) | a, b ∈{the files related to Joe, external storage media devices, Windows Registry on Adam's laptop}}. To list all USB-based devices which have been connected to Adam's laptop, Moti uses the tool USBDeView to extract all USB device information from Windows Registry on Adam's laptop and retrieves 20 distinct USB devices.

Since moving to consideration of unknown USB devices has the possibility of leaving the case open-ended and taking the investigation into overtime, Moti decides to use our ranking scheme for the second time. Now he has two new hypotheses h_4 and h_5 as well as the three hypotheses h_2= "The emails from Joe are in Joe's laptop," h_3^1= "Adam erased the 20 emails by using Windows Mail," and h_3^2= "Adam erased the 20 emails by using Eraser" derived during the first working day. Including the information obtained in Rounds 4 and 5 (Table 3), Moti re-enters the hypotheses into our ranking analysis model and obtains the results shown in Table 3.

The two new hypotheses are ranked top in the "Overall" column in Table 2. Hypotheses h_5, h_4 and $h_3^2$are closely related because of their strong correlation in timeline. Additionally, a key factor likely affecting the sorted results is the USB device plugged in for 6 minutes before the laptop was shut down on October 1, 2009. More specifically, around 54GB data could be written to this USB thumb drive in 6 minutes at an average speed of 15MB/s. Additionally, the use of the laptop in the early morning breaks a habit of Adam's. All in all, Moti is confident that Adam moved the valuable evidence files to a USB device.

So, Moti only keeps the files related to Joe and the USB thumb drive firstly identified in Round 4 in the object set $\mathbf{O}_5$= {the files related to Joe, USB thumb drive}; and the relational class becomes $\mathbf{R}_5$= {(a,b) | a, b ∈{ the files related to Joe, USB thumb drive}}. Then, Moti updates his hypothesis set to include h_5^1= "Adam moved the files related to Joe to a USB thumb drive."

Round 6: Since each USB device has a distinctive signature, after about 30 minutes, Moti is able to locate the exact device in the evidence locker. Having made a forensic copy of this USB drive, Moti browses its contents and finds a very large JPG file over 200 MB in size called "1.jpg" on the drive. Moti writes a new hypothesis h_6= "Adam hid the files related to Joe in the file 1.jpg." Moti has two elements in the object set $\mathbf{O}_6$= {the files related to Joe, 1.jpg} and a trivial relational class $\mathbf{R}_6$= {(a,b) | a ∈{the files related to Joe}, b ∈{1.jpg}}. To recover the hidden data, Moti runs the program Scalpel and recovers an encrypted RAR file from "1.jpg." Though the recovered RAR file is password protected, Moti is able to browse the names of the files and directories inside. For example, some of the file names contain the name of Joe's regular customer Wong, such as "transaction records with Wong," "check from Wong," "Wong's address" and so are clearly related to Joe's drug trafficking case.

It is almost the end of the second working day, and Moti decides to stop his investigation. Moti writes a case report illustrating his key steps in locating the documents embedded in a JPG file on a thumb drive. He suggests that the forensic lab should attempt to decrypt the contents of the recovered RAR file.

SUMMARY

In summary, this case study demonstrates the use of hypothesis generation and hypothesis ranking during a digital forensic investigation where many assumptions and decisions are made by investigators. In particular, we demonstrate an example of applying our new architecture to speed up the investigative process while retaining the benefits of using hypothesis generation and relationship building techniques. Our new approach moves beyond relationship building, already used by several authors, to focus on hypotheses generation and analysis. With the

Table 3. Output from the RAM after Round 5

Overall	$PM_2(h_2)$	$PM_3(h_3)$	$PM_4(h_4)$	$PM_5(h_5)$
h_5	h_2	h_3^1	h_4	h_5
h_4		h_3^2		
h_3^2				
h_3^1				
h_2				

use of a ranking scheme, the investigator can quickly identify those important relations and hypotheses about which he is confident and is able to eliminate less convincing statements.

REFERENCES

Al-Zaidy, R., Fung, B., Youssef, A. M., & Fortin, F. (2012). Mining criminal networks from unstructured text documents. *Digital Investigation*, *8*(3-4), 147–160. doi:10.1016/j.diin.2011.12.001

Batten, L. M., & Pan, L. (2008). Teaching digital forensics to undergraduate students. *IEEE Security & Privacy*, *6*(3), 54–56. doi:10.1109/MSP.2008.74

Batten, L. M., & Pan, L. (2011). Using relationship-building in event profiling for digital forensic investigations. In *Proceedings of the Third International Conference on Forensics in Telecommunications, Information, and Multimedia* (Vol. 56, pp. 40-52).

Carrier, B. D. (2006). *A hypothesis-based approach to digital forensic investigations* (CERIAS Tech. Rep. No. 2006-06). West Lafayette, IN: Center for Education and Research in Information Assurance and Security, Purdue University.

Chu-Carroll, J., Brown, E. W., Lally, A., & Murdock, J. W. (2012). Identifying implicit relationships. *IBM Journal of Research and Development*, *56*(3-4), 1–10.

Chu-Carroll, J., Fan, J., Boguraev, B. K., Carmel, D., Sheinwald, D., & Welty, C. (2012). Finding needles in the haystack: Search and candidate generation. *IBM Journal of Research and Development*, *56*(3-4), 1–12.

Chu-Carroll, J., Fan, J., Schlaefer, N., & Zadrozny, W. (2012). Textual resource acquisition and engineering. *IBM Journal of Research and Development*, *56*(3-4), 1–11.

Ferrucci, D. A. (2012). Introduction to "This is Watson." *IBM Journal of Research and Development*, *56*(3-4), 1–15.

Gondek, D. C., Lally, A., Kalyanpur, A., Murdock, J. W., Duboue, P. A., & Zhang, L. (2012). A framework for merging and ranking of answers in DeepQA. *IBM Journal of Research and Development*, *56*(3-4), 1–12.

Harabagiu, S., Hickl, A., & Lacatusu, F. (2006). Negation, contrast and contradiction in text processing. In *Proceedings of the 21st National Conference on Artificial Intelligence*, Boston, MA (pp. 755-762).

Herstein, I. N. (1975). *Topics in algebra*. New York. NY: John Wiley & Sons.

Jankun-Kelly, T. J., Wilson, D., Stamps, A. S., Franck, J., Carver, J., & Swan, J. E. (2009). A visual analytic framework for exploring relationships in textual contents of digital forensics evidence. In *Proceedings of the IEEE 6th International Workshop on Visualization for Cyber Security*, Atlantic City, NJ (pp. 39-44).

Kalyanpur, A., Patwardhan, S., Boguraev, B. K., Lally, A., & Chu-Carroll, J. (2012). Fact-based question decomposition in DeepQA. *IBM Journal of Research and Development*, *56*(3-4), 1–11.

Li, H. (2011). A short introduction to learning to rank. *IEICE Transactions on Information and Systems*, *94*(10), 1854–1862. doi:10.1587/transinf.E94.D.1854

Marrington, A., Mohay, G., Morarji, H., & Clark, A. (2010). A model for computer profiling. In *Proceedings of the IEEE International Conference on Availability, Reliability, and Security*, Krakow, Poland (pp. 635-640).

Marrington, A. D. (2009). *Computer profiling for forensic purposes* (Unpublished doctoral dissertation). Queensland University of Technology, Queensland, Australia.

Murdock, J. W., Fan, J., Lally, A., Shima, H., & Boguraev, B. K. (2012). Textual evidence gathering and analysis. *IBM Journal of Research and Development*, *56*(3-4), 1–14.

Pan, L., Khan, N., & Batten, L. M. (2012, June 6-8). Using hypothesis generation in event profiling for digital forensic investigations. In *Proceedings of the 7th International Workshop on Digital Forensics & Incident Analysis*, Crete, Greece (pp. 76-86).

Prager, J., Duboue, P., & Chu-Carroll, J. (2006). Improving QA accuracy by question inversion. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia (pp. 1073-1080).

Radev, D. R., Prager, J., & Samn, V. (2000). Ranking suspected answers to natural language questions using predictive annotation. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, Seattle, WA (pp. 150-157).

Wang, C., Kalyanpur, A., Fan, J., Boguraev, B. K., & Gondek, D. C. (2012). Relation extraction and scoring in DeepQA. *IBM Journal of Research and Development*, *56*(3-4), 1–12.

Willassen, S. (2008). Hypothesis-based investigation of digital timestamps. In Ray, I., & Shenoi, S. (Eds.), *IFIP International Federation for Information Processing, Advances in digital forensics IV* (Vol. 285, pp. 75–86). New York, NY: Springer.

Lynn Batten holds the Research Chair in Mathematics at Deakin University (Melbourne, Australia), and is the director at Deakin of the Research Laboratory "Securing Cyberspace." Professor Batten specializes in research in information security and privacy across a spectrum of areas including malicious software, digital forensics and cryptographic protocols. She is a Fellow of the Australian Computer Society, a Senior Member of the IEEE and a Graduate of the Australian Institute of Company Directors.

Lei Pan is a lecturer at Deakin University. He mainly works on digital forensics and computer security. He became a staff member at Deakin University in 2008 after he obtained his PhD degree from Deakin University in testing digital forensic tools; his thesis was supervised by Professor Lynn Batten.

Nisar Khan completed a Master's Degree in Information Technology at University Putra, Malaysia and a Master's degree in Statistics from IUB Pakistan; he is currently working on a PhD at Deakin University Australia. Mr. Khan worked as a lecturer for ten years at several institutes of higher learning in Malaysia and the Middle East and at the same time provided consultancy to industry and academia. His areas of expertise are: Teaching blended with contemporary E-Learning platforms, Research, New Program Development, and Management. His career objectives are to contribute strong technical skills, excel in innovative technology research and development as a highly experienced and qualified academician. His current research is in digital forensics investigation.