Benchmarking search engines on forensic data

Joachim Hansen

NTNU
Norwegian University of
Science and Technology

Master's Thesis Project Description - IMT4601
NTNU in Gjøvik, 2017

Department of Computer Science and Media Technology
NTNU in Gjøvik
PO box 191
NO-2802 Gjøvik, Norway

Abstract

This paper is a master thesis proposal written for the course IMT4601 Research project planning. The proposal aims at evaluating the performance of search engines and search engine functionality on forensic data. The framework for evaluating search engines can give forensic practitioners the information they need to determine if search engines is applicable as a tool in the digital forensic investigation. Experiments is proposed to measure the time complexity, storage complexity, recall, precision and F-measure for a selection of search engines, search engine functionality and indexing strategies. A literature review looks at how search can be applied to the digital forensic investigation, which search engines are out in the wild, the search capabilities of the search engines, the utility of the search engines search functionality and the importance of recall and precision. The proposal also includes information like how sources was obtained, a Gant chart that shows the activities and milestones, a feasibility study etc.

Contents

Glossary

AF	Anti Forensics
DFI	Digital Forensic Investigation
FP	Forensic Practitioner
IR	Information Retrieval
TC	Time Complexity
MC	Memory Complexity
MVT	Memory Visualisation Tool
RBT	Red Black Tree
SE	Search Engine

1 Introduction

The purpose of this chapter is to present the reader with the topic of the thesis, the problem description, justification for doing the research, the research questions that will guide the research and the planned contributions of the research.

1.1 Topic covered by the project

Digital forensics investigations have to deal with a digital landscape where the amount of data increases in volume each year[1]. The big data problem introduces problems such as how can forensic practitioners (FP) process the data collected in their investigation in a reasonable amount of time and figuring out how to best handle the storage requirement of the data. Using relational databases to process the data is not appropriate as large portion of the data is unstructured[2].

Information retrieval systems like search engines (SE) have been used to help locate enterprise data. SE used in enterprises also have to deal with large volumes of heterogeneous data[3].

This master thesis proposal aims to evaluate the performance of search engines and search engine functionality on forensic data.

1.2 Keywords

Digital forensics, search engines, benchmarking, open source, recall and precision

1.3 Problem description

Forensic practitioners in digital forensics have to process large quantitative of structured and unstructured data. The processing of data have to be reliable, forensically sound and preferably be solved with a low memory and time complexity. Forensic practitioners can use one of many Search Engines (SE) to aid them on this task.

By knowing which SE that is out there, which algorithms they use and their performance, then the forensic practitioners can make a conscious decision on which SE that best aid them on the forensic process.

1.4 Justification, Motivation and benefits

Digital forensic investigation have a big data problem. Without tools that can search the data within a small time frame and provide relevant hits, then forensic investigation cannot examine the evidence in a timely manner. Which in turn can negatively effect the justice system capability of convicting criminals.

The ability for search engines to process large amount of forensic data will be benchmarked in this paper. The benchmark can provide relevant information needed to determine whether or not to invest resources on integrating search engines into the digital forensic investigation process.

1.5 Research question

Table 1: Research questions

Research question 1	
Which index strategy leads to best performance for the search engine	
Variable	Time complexity, storage/memory complexity, recall, precision, F measure
Group	Selection of SE, SE functionality, forensic data,index strategies

Research question 2	
How well does search engine functionality perform on forensic data?	
Variable	Time complexity, storage/memory complexity, recall, precision, F measure
Group	Selection of SE, SE functionality, forensic data

Research question 3	
How well does search engines perform on forensic data?	
Variable	Time complexity, storage/memory complexity, recall, precision, F measure
Group	Selection of SE, SE functionality, forensic data

In table 1, 3 different research question is presented with the variables and groups to be tested.

1.6 Planned contributions

The contribution is a framework to evaluate the performance of a selection of search engines and their search functionality in the domain of digital forensics.

2 Related work

In regards to the problem description it was unfortunately infeasible to obtain information on low level algorithms used by the search engines in table 2. This is because this information was not made accessible in the software documentation and undergoing a source code inspection would be too time consuming. But information on the search engines search functionality was possible to find on the search engine website and documentation pages. This trusted information sources will be later verified in search engines used in the master thesis experiment. The research questions is closely linked to the experiments. This chapter includes independent study into search engines found in the wild, their search capabilities and search utilities. This is information is necessary in order to begin to answer the research questions in the master thesis.

The literature review is divided up in the following subsections:

1. Application of search in digital forensic investigation: A review on the literature for the last 5 years on how search can be applied to digital forensic investigations. This section is further divided into collection, examination and analysis. Which are phases in the digital forensic investigation process model discussed in [4].
2. Search engines: A overview of the search capabilities for a number of search engines that are open source, recently in development and that are not primarily web search engines.
3. Search utility: A look into the utility of the search engines search functionality.
4. How search engines should perform in a digital forensic domain

2.0.1 Application of search in digital forensic investigation

Collection phase

Privacy law can regulate what method FP can use when collecting evidence. One paper [5] created a privacy protected scheme, where FP can perform a keyword search on encrypted emails. The individual emails could only be decrypted if the amount of exact matching non-blacklisted keywords provided by the FP are equal or above a certain threshold. Blacklisting or whitelisting certain keywords can make it harder for an attacker to perform a dictionary attack.

The paper by [6] argued that volume information found in the open source distributed file system platform XtreemFS is of interest to FP. The information can be used to search to find particular volumes of interest and the size of the volumes to determine if acquisition is practical. FP can search for the string "xtreemfs@" to find out if a node is connected to XtreemFS.

Examination phase

It was claimed in [7] that it is commonplace for Forensics Practitioners (FP) to maintain a database of hashes of know illegal images and videos. FP can hash media collected in a investigation and search the database for matches. This approach has obvious limitation against anti forensics (AF) approaches such as resizing of the images. To improve upon this scheme the paper creates a custom database called hashdb, that stores hashes of the individual data blocks of files. This solution is more resistance against small file modific-

ation, as many of the data blocks would remain unchanged. Searching the database for matches of crime media can return a single match or a candidate list.

While not being widely adopted by the digital forensics community, approximate matching can be used to detect semantically and syntactical similar files and match it against a reference dataset[8]. Semantically similar files are files such as images that look alike in the eyes of humans. For example otherwise identical images, one in white and black and the other in colour would be perceptually the same file. The application of searching for semantically similar files can aid FP to find the origin of files of interest. Syntactical similar files are files that look similar on the byte level. Approximate hash based matching (AHBM) is not appropriate for images as they can look the same, but have different encodings. But are well suited for dealing with unstructured data such as text files, memory dumps and fragmented files. The paper concludes that the same results can be accomplished with string search as with approximate matching, but this would require far more from the FP.

One issue with collected forensic image of a storage device like hard disk drive (HDD) is duplicated files[9]. Processing duplicated files leads to unnecessary overhead in the examination phase. One way to solve this issue is by arranging the files in a red black tree structure (RBT). Duplicate nodes in this structure can be found by searching using wildcards. After identifying duplicate nodes their child nodes will be rearranged in the tree and then the duplicate node will be removed from the structure. The time complexity for searching, inserting and removing nodes in RBT is $O(\log_2(n))$ for the average and worst case. This proposed solution do not state in detail how their scheme identifies files with the same content. While identifying the same file names using wildcard seems resonable, hash matching is more appropriate for telling if two files have identical content.

A proposal was made in [10] to identify duplicate images where the file name, file extension or file attributes (e.g hidden, compressed, encrypted and protected Operating System File) did not match the source image. The proposal used the source modified timestamp to search for duplicate files. 1000 files spread across 30 folders totalling 3.09 GB in size was processed in 1 minute and 32 seconds. The same files spread across 300 folders took 16 minutes 23 seconds longer to process. So its application is limited to environments with a small number of folders. The proposal is also vulnerable to tampering done to the modified timestamp attribute.

According to [11] the United State Supreme Court are beginning to demand that the examination process are limited in its scope. This means that the goals and objectives must be clearly stated, as well as a justification for what the examiner will search for and the boundary of the search. Failure to comply could negatively effect their case in court. This restriction might force a better resource management of the examiner resources. But it can also make it more difficult to examine evidence that is hidden in unusual locations, as its examination would be difficult to justify. Simply searching for everything in a Gigabytes or Terabytes search space would not solve the problem as this task is infeasible even when using common digital forensics tools or automated tools[12, 13]. The courts also put constraints on how long seized data can be processed by the examiner, before it is returned to its owner[14]. It is argued in [15] that the searching by the examiner, can be aborted after the most probable places have been processed. More specific search criteria can reduce privacy violations and reduce number of false positive hits[16]. The question then arises how specific can you be before negativity impacting

the recall rate.

One study [17] showed that usernames and passwords found in computer memory can be used to identify which websites the credentials belongs to. A search condition like “&Email” and “&Passwd” can be used to search for usernames and passwords in memory. Some usernames and passwords that belongs to particular websistes can be retrived with a unique search pattern, others can be found by using the same search condition. The non-unique search conditions can use the session component found in memory to uniquely identify the website. Having a reference database for this mapping can be useful for forensics examiners that want to understand suspect activity online. Maintaining the referance database beyond the most common websites would be impractical.

Email spam folders are often overlooked by FP as they mostly consist of junk[18]. Criminals can craft their messages in such a way that it will to be picked up by the spam filter and hide their activities from law enforcement. Keyword searches and manual review of the spam emails is therefore important to find obfuscated evidence. The folder could be a way for criminals to obfuscate their activities, and should therefore be searched.

FP have to search though large volumes of heterogeneous data. One study[19] evaluated the performance of clustering techniques on a forensic dataset containing 2640681 search hits. They achieved a precision improvement of a factor 15 over non-clustering and a overall average precision of 67%.

One paper[20] created a search algorithm called ScalClone that aims to find exact and inexact code fragments between analysed and un-analyzed malicious assembly files. Exact fragments are identified by searching for regions with the same hash value. Inexact fragments are fragments that share many mnemonics and operand types. They are identified by first constructing a binary vector with respect to feature frequency and features mean value, and then comparing the co-occurrences of the fragments. If the co-occurrences count is greater or equal to the similarity threshold, then the fragment is considered a inexact clone. Inexact search is not effected by reordering as the frequency of the mnemonics remains unchanged. Obfuscation by adding do-nothing instruction drops the recall rate to 90% and compiler optimization drops it to 62%.

A survey [21] stated that string search in volatile memory examination is useful in order to find residue of user activity, passwords, encryption keys and side effects of malicious scripts. Searching in swapped out memory pages in windows can potentially provide evidence of old user activity, as the swapped file is often not cleared after system reboot[22]. Another study [23] showed how searching for the string ’for deletion’ in a Hadoop Distributed File System (HDFS) is useful to find evidence of deleted files. The paper [24] claimed that only the row directory is overwritten with a NULL value when a row is deleted in the database DB2 or SQL sever. This allows a FP to search these databases for the deleted rows and restore them by considering the valid row directory values of their previous and following row directory entry.

Pool tag scanning is a type of exhaustive search on volatile memory that is used to find data structures such as direct kernel object manipulation (DKOM) which is used by malware to hide processes[25]. The study [25] stated that exhaustive search might not be appropriate for time sensitive investigations. They therefore created pool tag quick scanning, which reduces the search space to memory pages related to pool allocations. The search space reduction can be "multiple orders of magnitude" and the accuracy of

the search results remains high.

A comparison was done in [26] to test the accuracy and speed of which experienced participants in networking, windows operating system, malware and incident response, are to solve forensics tasks. The participants where given the same tasks and the same forensics image. They where split into two groups, one that used normal text search and the other that searched using a memory visualization tool (MVT). The MVT showed relationships between the data and had a whitelisting algorithm that removed known good files from the search space. The results showed that the participants that used the MVT completed the tasks faster and more accurate. I infer from the text that the number of participants are 10 (minus one outlier). Laying to much weight from the results on this low sample size might not be appropriate.

The study [27] compared the state of the system before and after forensics examination using the following bootable forensics environments: Knoppix v7.0, Helix 3 Pro 2009R3 and Kali Linux v1.0. Keyword searches was used during the examination process to simulate an investigation. The hash value taken on the forensics image before and after examination, did not match in any case. It was mainly the “last accessed” timestamps on files that was altered after the examination. Performing keyword searches in those environments can therefore be problematic in cases where establishing a timeline is important.

It is argued in [28] that keyword searches resulting in large number of false positive hits, can be reduced by using background knowledge from the investigation. Fuzzy logic can also be applied to find elements missed by the normal keyword search such as misspelled words and slang terms. While keyword search algorithm are useful, they are inept at processing terabytes of data[12].

One study [13] used keywords search terms to cluster forensic data to reduce examination overhead. There is one cluster per search term. In order to help the examiner choose good search terms, the system returns the most frequent used search terms found in the forensics data. Both with and without suggestions, the system performs good with respect to average precision and recall. The system is also scalable as the runtime grows linearly with the number of documents.

Analysis

Finding evidence of deletion of user activity on the suspect machine is of interest of FP[29]. Searching the Update Sequence Number (USN) Journal file on the NTFS can reveal when and where files have been created, viewed, renamed, moved or deleted.

One study [30] mined 1100 chat logs to find the most significant terms, users and chat sessions. Two bigraphs are constructed. The mapping in the first bigraph is such that we can observe which term (Hub) has been said by which users (Authorities) and what terms (Hubs) have been said by a user (Authority). The second bigraph has similar mapping, but the Hub is the term and the authority is the chat session. A self-customized hyperlink-induced topic search (HITS) algorithm is used to iteratively set the Authority and Hub score. A selection of the highest scoring users, chat sessions and terms are used together with user metadata and session metadata to construct a social graph. Clustering is applied on the social graph to find shared interest and interactions between users.

One study [31] showed how traces found from volatile memory in IEEE 802.11 wireless devices, that is in radio range from each other can answer important forensics questions like Who, When and Where. There are two types of broadcast traffic frames that

can answer these question. As their format is known, they can easily be found by using regular expression search. The probability that the frames are still in the devices volatile memory depends on external and internal conditions like the extent and nature of the broadcast traffic processed by the device and the configurations of the device. This methodology would therefore only work in a few real life scenarios and mostly in non-urban areas.

Search helps file carving tools identify header, footer and fragments used to identify where a file begins and end and use this information to restore the file [32]. Some file carving tools are able to restore files independent on the underlying file system. Exhaustive search can be used to find each combination of header and footer of a video and then try to validate/decode on the restored file to see if it is a valid video. Search can be used to find the order of the fragments and codecs search codes to identify fragments belonging to videos.

The FP may encounter digital environments where the binary data is encoded using multiple different UNICODE encodings and that the type of UNICODE are unknown[33]. The share number of possible UNICODE encodings means that the same text can be represented in many different ways. Resolving the underlying encoding in the worst case can require number of search passes equal to the number of possible encodings. The average case is much better as many encodings are not widely used. The regular expression search engine lightgrep aims to deal with the encoding problem. Lightgrep uses UNICODE characters as string literals in the regex expression to be encoding independent. For handling the encoding Lightgrep uses multi pattern search enabling it to search for multiple encodings in parallel. The search engine currently support 180 encodings making it possible to perform UNICODE-aware searches.

2.0.2 Search engines

- S_1 = Full text search
- S_2 = Faceted search
- S_3 = Spatial/Geospatial search
- S_4 = Fuzzy search
- S_5 = Streamed search
- S_6 = Phonetic search
- S_7 = Semantic search

Table 2: Open source desktop/intranet search engines and their default search capabilities
Source: [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52], [53], [54], [55], [56], [57], [58], [59], [60], [61] [62], [63]

Name	S_1	S_2	S_3	S_4	S_5	S_6	S_7	Update
Dezi	✓	✓						28.11.2016
Apache Solr	✓	✓	✓	✓	✓			06.03.2017
Sphinx	✓			✓				08.09.2016
Sifaka	✓							25.01.2017
OpenSearchServer	✓	✓	✓			✓		13.01.2017
Luwak					✓			06.03.2017
Datafari	✓	✓						23.03.2017
Elasticsearch	✓	✓	✓	✓		✓		24.04.2017
groonga	✓		✓					24.04.2017
tantivy	✓							23.04.2017
tntsearch	✓		✓	✓				20.04.2017
pouchdb-quick-search	✓							22.02.2017
OpenSemanticSearch	✓	✓		✓			✓	16.04.2017

2.0.3 Search utility

According to the whitepapers[64],[65] Full text search (FTS) is suitable for finding relevant documents in a large set of unstructured data. A lot of the data gathered in a forensic investigation is unstructured[66]. It is more appropriate to use FTS to respond to ad hoc request than requests with a predefined answer[64]. A document in FTS is considered a list of searchable terms (e.g. words and numbers)[65]. The terms are usually indexed in order to make them easier to search.

Faceted search is a way of traversing the corpus based on categories (facet) and subcategories (facet values)[67]. In faceted search it is possible to find the same the same data points by using different traversal paths. Faceted search is useful for exploring the corpus and the facet values aid the searcher to create more precise search phrases. It is common practice in faceted search systems that only the most frequent facet values are shown. This makes finding more obscure items difficult.

Fuzzy keyword search retrieves both documents that matches exactly with the search phrase and those within a similar distance[68]. The distance can be measured by using the Levenshtein distance. Which compares the minimum number of insertions, deletions or substitutions are needed for string A to equal string B. The paper [69] claims that fuzzy search is helpful when the searcher have do not have sufficient domain knowledge of the dataset he is searching.

Phonetic search is matching based on similar sounding words[70],[55]. One example of a phonetic algorithm is Soundex. It encodes a word into a 4 character code starting with the same character as the word[70]. Similar sounding characters like s,f,p and v are represented by the same number. Repeating characters, vowels and certain letters are ignored by the algorithm. Truncation and padding are used to make sure that all words are represented by a 4 character code. The limitation with this approach is that only words starting with the same letter would have a chance to match with the same code. Phonetic algorithms are designed to handle specific languages, making them limited in their utility[55]. The aim of Phonetic search is not improving precision but to increase the recall rate.

Geospatial search is searching a corpus where the documents have associated geographic data such as latitude and longitude. One example of using the location data is to search for registered criminals that lived in the vicinity of a crime scene[71]. It can also be used to find all previous search warrants on a address or all search warrants in some proximity to a given address.

Documents that do not contain the terms of the user query can still be relevant[72]. Classical retrieval based on lexicographic term matching will not retrieve documents that are lexicographically different but semantically similar. To improve information retrieval of documents Semantic search can find semantically similar terms that are often overlooked by using stemmed synonyms or Ontology. Ontology models a domain into concepts, attributes and relations[73].This model provides the semantic reasoning needed to retrieve meaningful documents with respect to the user query[74].

Streamed search was explained in [75, 76]. In traditional full text the documents are often indexed using inverted indexing to optimize the time it takes to find the queried documents. Running all possible queries on the documents works well if the complexity of the queries and the data velocity is low. Network log files is a example of a stream (continuous data flow) where traditional search is impractical. Stream search uses inverted indexes on queries instead of documents. By doing so it is possible to take the new log entry and query the inverted index to see which indexed queries match the new entry. Now the search have identified the minimum number of queries that need to run on the new entry. This approach could potentially save high amount of computer resources.

2.0.4 How search engines should perform in a digital forensic domain

The importance of the measurements precision and recall in Information Retrieval (IR) systems, like Search Engine (SE) depends on the application[77]. In the domain of Digital Forensic Investigation (DFI) precision is more important in the early phases of the forensic investigation, as relevant evidence is vital to guide the process of finding new evidence. At the later stages of the DFI, recall becomes more significant than precision, as Forensic Practitioner (FP) wants all available evidence to build a court case.

2.1 Handling problems

The table 3 show which search phrases and search resources used to collect the sources and the number of resulting hits. Other sources like finding the search engines web pages was found in a snowball fashion. Where relevant project links on sourceforge and github was used to locate the search engines.

Table 3: How the sources was located

Query	Search resource	Hits
("Abstract":enterprise AND search AND engine), Year: 2014-2017	http://ieeexplore.ieee.org	27
"Enterprise search", in abstract, year: 2014-2017, source type: Scholarly Journals	http://search.proquest.com/abicomplete/	24
recordAbstract:(+enterprise +search) , year: 2014-2017	http://dl.acm.org	44
in abstract (Solr OR Elastic-Search)	https://arxiv.org	6
in abstract, title and keywords: Enterprise search, year 2014-2017	http://www.sciencedirect.com	47
in abstract:Solr OR Elastic-Search, Scholary journals, year:2014-2017	http://search.proquest.com/	47
in abstract (Solr OR Elastic-Search), year:2014-2017	http://ieeexplore.ieee.org	72
recordAbstract:(Solr Elastic-search), year:2014-2017	http://dl.acm.org/	18
information retrieval unstructured data (general search), year:2014-2017	http://ieeexplore.ieee.org	122
"information retrieval" "unstructured data" survey	https://scholar.google.no/	2990
(+"Digital forensics" +search) - any fields	http://dl.acm.org	7
(+"Computer forensics" +search) - any fields	http://dl.acm.org	7
in journal "Digital Investigation" : search, 2014-2017	http://www.sciencedirect.com	161
in book "Digital Forensics Threatscape and Best Practices" - year: 2016 - search phrase: search	http://www.sciencedirect.com	10
in publication "IEEE Transactions on Information Forensics and Security", year:2014-2017	http://ieeexplore.ieee.org	42
basic search " Digital forensics search", year:2014-2017	http://ieeexplore.ieee.org	65

3 Choice of methods

The master thesis will use quantitative methodologies in order to answer the research questions. I would first need to select which search engines to test in my experiment. Then I would need to choose which subset of the search engines filters/search functionality to include in the experiment. Then the selected search engines and search functionality have to be setup/implemented on the test environment. A large forensic dataset have to be acquired, so that the experiment will be run with realistic data types and volume.

To answer the first research question I would need to understand how to implement various indexing strategies. And then test how well these perform in the different search engines. All research questions will require implementation of search engines and search functionality to be benchmarked in the test environment.

The list below is the proposed methodology of how to collect data on recall, precision, F-measurement, time complexity and memory complexity for the experiment. These steps are inspired from the paper [78].

1. A query in one form or another (e.g. filter) will be created using search engine X and search functionality Y, to find some relevant data in the forensic dataset
2. Based on the query and domain knowledge of the dataset, on or more people will decide which documents/data are relevant before the execution of the query statement.
3. Execute the query in the SE (start the search). At this step Memory Complexity (MC) and Time Complexity (TC) should be measured of the algorithms. One possible way to measure this is checking the resource management system on the test environment.
4. Based on the number of actual retrieved documents/data and the number of relevant documents/data we can calculate recall, precision and F-measurement.

collecting these data points should be plausible, as information retrieval systems are often evaluated by the recall and precision metrics. And memory and time complexity of the running process are often tracked by the computer operating system.

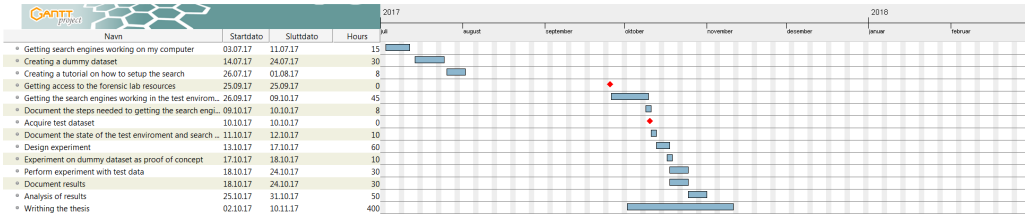
For all experiments I need to setup a test plan that need to contain:

- The state of the program (what configurations have been made)
- The scope of the test
- Which tools are used to get the measurements
- configurations on the testing environment
- What keywords are to be used
- Description of the nature of the dataset
- Have test that can play on the strength and weakness for different search functionality.
- Describe the indexing strategy used in the test
- Naming the search engines and search functionality used in the test.
- Verify that the search engine has indeed the advertised search functionality, by using black box testing.

3.1 Milestones, deliverables and resources

In figure 1 you can see the activities and milestones. The activities can be found under 'navn' and begin date and end date can be found under 'startdato' and 'sluttdato' respectively. The red icon represent a significant event (milestone) in the project. The number of man hours needed to complete an activity can be found under 'hours'. From the chart I can see that the sooner I get access to the forensic resources and test data set, the better. This is because many of the activities depend on them in order progress.

Figure 1: Gant chart (need to zoom in 400%)



The deliverables:

- Introduction
- Theory contents
- Description of dataset
- Description on experimental design
- Proof of concept with dummy dataset
- Results section
- Discussion section
- Conclusion
- Abstract

3.2 Feasibility study

Taking measurements for recall, precision and f-measure for a information retrieval system was done in [78].

Using the documentation and source code for the open source search engine under inspection, it will be easier to understand how to best measure recall, precision, f-measure, time complexity and storage complexity. Data on time and memory complexity can possible also be collected by the resource management system running on the experiment computer environment.

3.3 Risk analysis

The table 4 is used to reference how severe a risk is with respect to impact and likelihood. The colour red indicates that the risk have to be reduced. With yellow the risk should be reduced. And green is the acceptable level of risk. Below the table I have made a list of the 5 most significant risk elements in my thesis.

Table 4: Risk Table

Impact / Likelihood	Very Unlikely	Remote	Seldom	Probable	Frequent	Very Frequent
Severe						
Significant						
High						
Moderate						
Low						
Minimal						

- Not acquiring the forensic dataset needed for the experiment. Katrin Franke said that the forensic lab could obtain the forensic samples for me. But in case they fail coming though with that in the early stages of my thesis, then I should create a backup dataset.
- I would need access to some resources in the forensic lab. To minimize the risk of not getting these resources, I should get a written agreement with key players in the forensic lab and have close communication.
- A lot of time might be needed to familiarize myself with the different search engines in order to create my experiment. If this overhead is overwhelming, then it could negatively impact the thesis. I could spend some time in the summer vacation to test these search engines
- It takes some time before I get the forensic dataset needed to perform any experiment. A solution to this problem can be to have a small dummy dataset for creating a proof of concept. I would still need the larger dataset, but the dummy dataset would allow me to progress.
- Loosing time due to sick days. The best way to avoid that sick days effect the thesis is planning and starting working early.

3.4 Ethical and legal considerations

There are 3 legal considerations:

- The benchmark experiment can only be performed on search engines with licences that allows benchmarking. This can be managed by only selecting those search engines where benchmarking is allowed.
- The nature of the forensic dataset. The dataset should not contain information that is illegal to store.
- Compliance with written or verbal contracts/agreements with how the forensic lab resources used in the thesis should be handled.

Bibliography

[1] Zawoad S, Hasan R. Digital Forensics in the Age of Big Data: Challenges, Approaches, and Opportunities. In: 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems; 2015. p. 1320–1325.

[2] Yafooz WMS, Abidin SZZ, Omar N, Idrus Z. Managing unstructured data in relational databases. In: 2013 IEEE Conference on Systems, Process Control (ICSPC); 2013. p. 198–203.

[3] Li Y, Liu Z, Zhu H. Enterprise Search in the Big Data Era: Recent Developments and Open Challenges. Proc VLDB Endow. 2014 Aug;7(13):1717–1718. Available from: http://dx.doi.org/10.14778/2733004.2733071.

[4] Palmer G, Corporation M. A Road Map for Digital Forensic Research; 2001. Accessed 29.04.17. Available from: http://dfrws.org/sites/default/files/session-files/a_road_map_for_digital_forensic_research.pdf.

[5] Armknecht F, Dewald A. Privacy-preserving email forensics. Digital Investigation. 2015;14, Supplement 1:S127 – S136. The Proceedings of the Fifteenth Annual {DFRWS} Conference. Available from: http://www.sciencedirect.com/science/article/pii/S1742287615000481.

[6] Martini B, Choo KKR. Distributed filesystem forensics: XtreemFS as a case study. Digital Investigation. 2014;11(4):295 – 313. Available from: http://www.sciencedirect.com/science/article/pii/S1742287614000942.

[7] Garfinkel SL, McCarrin M. Hash-based carving: Searching media for complete files and file fragments with sector hashing and hashdb. Digital Investigation. 2015;14, Supplement 1:S95 – S105. The Proceedings of the Fifteenth Annual {DFRWS} Conference. Available from: http://www.sciencedirect.com/science/article/pii/S1742287615000468.

[8] Bjelland PC, Franke K, Årnes A. Practical use of Approximate Hash Based Matching in digital investigations. Digital Investigation. 2014;11, Supplement 1:S18 – S26. Proceedings of the First Annual {DFRWS} Europe. Available from: http://www.sciencedirect.com/science/article/pii/S1742287614000085.

[9] Wang WB, Huang ML, Lu L, Zhang J. Improving Performance of Forensics Investigation with Parallel Coordinates Visual Analytics. In: 2014 IEEE 17th International Conference on Computational Science and Engineering; 2014. p. 1838–1843.

[10] Sharif SA, Ali MA, Reqabi NA, Iqbal F, Baker T, Marrington A. Magec: An Image Searching Tool for Detecting Forged Images in Forensic Investigation. In: 2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS); 2016. p. 1–6.

[11] Pollitt M. Chapter 2 - The key to forensic success: examination planning is a key determinant of efficient and effective digital forensics. In: Sammons J, editor. Digital Forensics. Boston: Syngress; 2016. p. 27 – 43. Available from: http://www.sciencedirect.com/science/article/pii/B9780128045268000022.

[12] Rogers MK. Chapter 3 - Psychological profiling as an investigative tool for digital forensics. In: Sammons J, editor. Digital Forensics. Boston: Syngress; 2016. p. 45 – 58. Available from: http://www.sciencedirect.com/science/article/pii/B9780128045268000034.

[13] Mascarnes S, Lopes P, Sakhare P. Search model for searching the evidence in digital forensic analysis. In: 2015 International Conference on Green Computing and Internet of Things (ICGCIoT); 2015. p. 1353–1358.

[14] Pollitt MM. Triage: A practical solution or admission of failure. Digital Investigation. 2013;10(2):87 – 88. Triage in Digital Forensics. Available from: http://www.sciencedirect.com/science/article/pii/S1742287613000030.

[15] Overill RE, Silomon JAM, Roscoe KA. Triage template pipelines in digital forensic investigations. Digital Investigation. 2013;10(2):168 – 174. Triage in Digital Forensics. Available from: http://www.sciencedirect.com/science/article/pii/S1742287613000261.

[16] Moser A, Cohen MI. Hunting in the enterprise: Forensic triage and incident response. Digital Investigation. 2013;10(2):89 – 98. Triage in Digital Forensics. Available from: http://www.sciencedirect.com/science/article/pii/S1742287613000285.

[17] Thongjul S, Tritilanunt S. Analyzing and searching process of internet username and password stored in Random Access Memory (RAM). In: 2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE); 2015. p. 257–262.

[18] Yu S. Covert communication by means of email spam: A challenge for digital investigation. Digital Investigation. 2015;13:72 – 79. Available from: http://www.sciencedirect.com/science/article/pii/S1742287615000432.

[19] Beebe NL, Liu L. Clustering digital forensic string search output. Digital Investigation. 2014;11(4):314 – 322. Available from: http://www.sciencedirect.com/science/article/pii/S1742287614001108.

[20] Farhadi MR, Fung BCM, Fung YB, Charland P, Preda S, Debbabi M. Scalable code clone search for malware analysis. Digital Investigation. 2015;15:46 – 60. Special Issue: Big Data and Intelligent Data Analysis. Available from: http://www.sciencedirect.com/science/article/pii/S1742287615000705.

[21] Case A, III GGR. Memory forensics: The path forward. Digital Investigation. 2017;20:23 – 33. Special Issue on Volatile Memory Analysis. Available from: http://www.sciencedirect.com/science/article/pii/S1742287616301529.

[22] III GGR, Case A. In lieu of swap: Analyzing compressed {RAM} in Mac {OS} X and Linux. Digital Investigation. 2014;11, Supplement 2:S3 – S12. Fourteenth Annual {DFRWS} Conference. Available from: http://www.sciencedirect.com/science/article/pii/S1742287614000541.

[23] Leimich P, Harrison J, Buchanan WJ. A {RAM} triage methodology for Hadoop {HDFS} forensics. Digital Investigation. 2016;18:96 – 109. Available from: http://www.sciencedirect.com/science/article/pii/S1742287616300780.

[24] Wagner J, Rasin A, Grier J. Database image content explorer: Carving data that does not officially exist. Digital Investigation. 2016;18, Supplement:S97 – S107. Available from: http://www.sciencedirect.com/science/article/pii/S1742287616300500.

[25] Sylve JT, Marziale V, III GGR. Pool tag quick scanning for windows memory analysis. Digital Investigation. 2016;16, Supplement:S25 – S32. {DFRWS} 2016 EuropeProceedings of the Third Annual {DFRWS} Europe. Available from: http://www.sciencedirect.com/science/article/pii/S1742287616000062.

[26] Lapso JA, Peterson GL, Okolica JS. Whitelisting system state in windows forensic memory visualizations. Digital Investigation. 2017;20:2 – 15. Special Issue on Volatile Memory Analysis. Available from: http://www.sciencedirect.com/science/article/pii/S1742287616301438.

[27] Mohamed AFAL, Marrington A, Iqbal F, Baggili I. Testing the forensic soundness of forensic examination environments on bootable media. Digital Investigation. 2014;11, Supplement 2:S22 – S29. Fourteenth Annual {DFRWS} Conference. Available from: http://www.sciencedirect.com/science/article/pii/S1742287614000589.

[28] Attoe R. Chapter 6 - Digital forensics in an eDiscovery world. In: Sammons J, editor. Digital Forensics. Boston: Syngress; 2016. p. 85 – 98. Available from: http://www.sciencedirect.com/science/article/pii/B978012804526800006X.

[29] Lees C. Determining removal of forensic artefacts using the {USN} change journal. Digital Investigation. 2013;10(4):300 – 310. Available from: http://www.sciencedirect.com/science/article/pii/S1742287613001084.

[30] Anwar T, Abulaish M. A social graph based text mining framework for chat log investigation. Digital Investigation. 2014;11(4):349 – 362. Available from: http://www.sciencedirect.com/science/article/pii/S1742287614001091.

[31] Minnaard W. Out of sight, but not out of mind: Traces of nearby devices' wireless transmissions in volatile memory. Digital Investigation. 2014;11, Supplement 1:S104 – S111. Proceedings of the First Annual {DFRWS} Europe. Available from: http://www.sciencedirect.com/science/article/pii/S1742287614000188.

[32] Mathew LM, R S, Kizhakkethottam JJ. A survey on different video restoration techniques. In: 2015 International Conference on Soft-Computing and Networks Security (ICSNS); 2015. p. 1–3.

[33] Stewart J, Uckelman J. Unicode search of dirty data, or: How I learned to stop worrying and love Unicode Technical Standard number 18. Digital Investigation. 2013;10, Supplement:S116 – S125. The Proceedings of the Thirteenth Annual {DFRWS} Conference13th Annual Digital Forensics Research Conference. Available from: http://www.sciencedirect.com/science/article/pii/S1742287613000595.

[34] karpet. karpet/Dezi; 2016. Accessed on 20.03.2017. Available from: https://github.com/karpet/Dezi.

[35] Karman P. Dezi::Config;. Accessed 23.04.17. Available from: https://metacpan.org/pod/Dezi::Config.

[36] Karman P. Dezi::Aggregator::DBI;. Accessed 23.04.17. Available from: https://metacpan.org/pod/Dezi::Aggregator::DBI.

[37] Apache. Index of /lucene/solr/6.4.2; 2017. Accessed on 22.03.2017. Available from: http://apache.uib.no/lucene/solr/6.4.2/.

[38] Targett C. Spatial Search; 2017. Accessed 23.04.17. Available from: https://cwiki.apache.org/confluence/display/solr/Spatial+Search.

[39] Bernstein J. Streaming Expressions; 2017. Accessed 23.04.17. Available from: http

s://cwiki.apache.org/confluence/display/solr/Streaming+Expressions.
[40] Targett C. Faceting; 2017. Accessed 23.04.17. Available from: https://cwiki.apache.org/confluence/display/solr/Faceting.
[41] Sphinx. Sphinx 2.3.2-beta downloads; 2016. Accessed on 22.03.2017. Available from: http://sphinxsearch.com/downloads/beta/.
[42] Sphinx. Sphinx 2.3.2-beta reference manual; 2016. Accessed 23.04.17. Available from: http://sphinxsearch.com/docs/devel.html#searching.
[43] Lemur. The Lemur Project; 2017. Accessed on 22.03.2017. Available from: https://sourceforge.net/projects/lemur/.
[44] lemur project. Sifaka; 2016. Accessed 23.04.17. Available from: http://www.lemurproject.org/sifaka.php.
[45] Opensearchserver. Configuring facets;. Accessed 24.04.17. Available from: http://www.opensearchserver.com/documentation/clients/php_client/facets.md.
[46] emmanuel keller. OpenSearchServer; 2017. Accessed on 22.03.2017. Available from: https://github.com/jaeksoft/opensearchserver.
[47] OpenSearchServer. Downloads and documentation; 2017. Accessed on 22.03.2017. Available from: http://www.opensearchserver.com/.
[48] romseygeek. flaxsearch/luwak; 2017. Accessed on 26.03.2017. Available from: https://github.com/flaxsearch/luwak.
[49] Julien. Advanced search feature (Datafari 3.2 and above); 2017. Accessed 24.04.17. Available from: https://datafari.atlassian.net/wiki/pages/viewpage.action?pageId=61282866.
[50] julienFL. francelabs/datafari; 2017. Accessed on 26.03.2017. Available from: https://github.com/francelabs/datafari.
[51] ElasticSearch. Full text search;. Accessed 24.04.17. Available from: https://www.elastic.co/guide/en/elasticsearch/guide/current/full-text-search.html.
[52] ElasticSearch. Facets;. Accessed 24.04.17. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/search-facets.html.
[53] ElasticSearch. Geo Distance query;. Accessed 24.04.17. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-geo-distance-query.html.
[54] ElasticSearch. Fuzzy query;. Accessed 24.04.17. Available from: https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-fuzzy-query.html.
[55] ElasticSearch. Phonetic Matching;. Accessed 24.04.17. Available from: https://www.elastic.co/guide/en/elasticsearch/guide/current/phonetic-matching.html.
[56] elasticsearch. elasticsearch; 2017. Accessed 24.04.17. Available from: https://github.com/elastic/elasticsearch.
[57] Groonga. Characteristics of Groonga; 2017. Accessed 24.04.17. Available from: http://groonga.org/docs/characteristic.html#groonga-overview.
[58] Groonga. The latest release; 2017. Accessed 24.04.17. Available from: http://groonga.org/.
[59] tantivy. tantivy; 2017. Accessed 24.04.17. Available from: https://github.com/tantivy-search/tantivy.

[60] TNTsearch. TNTsearch; 2017. Accessed 24.04.17. Available from: https://github.com/teamtnt/tntsearch.

[61] pouchdb-quick search. pouchdb-quick-search; 2017. Accessed 24.04.17. Available from: https://github.com/nolanlawson/pouchdb-quick-search.

[62] Search OS. Open Semantic Search;. Accessed 25.04.17. Available from: https://www.opensemanticsearch.org/.

[63] Search OS. Open Semantic Search; 2017. Accessed 25.04.17. Available from: https://github.com/opensemanticsearch/open-semantic-search-apps.

[64] Krellenstein M. Starting a Search Application; 2009. Accessed 25.04.17. Available from: https://whitepapers.em360tech.com/wp-content/files_mf/white_paper/lucid2.pdf.

[65] Lucidworks. Full Text Search Engines vs. DBMS (whitepaper);. Accessed 25.04.17. Available from: https://lucidworks.com/2009/09/02/full-text-search-engines-vs-dbms/.

[66] Guarino A. In: Reimer H, Pohlmann N, Schneider W, editors. Digital Forensics as a Big Data Challenge. Wiesbaden: Springer Fachmedien Wiesbaden; 2013. p. 197–203. Available from: http://dx.doi.org/10.1007/978-3-658-03371-2_17.

[67] Cleverley PH, Burnett S. Retrieving haystacks: a data driven information needs model for faceted search. Journal of Information Science. 2015;41(1):97–113. Available from: http://dx.doi.org/10.1177/0165551514554522.

[68] Li J, Wang Q, Wang C, Cao N, Ren K, Lou W. Fuzzy Keyword Search over Encrypted Data in Cloud Computing. In: 2010 Proceedings IEEE INFOCOM; 2010. p. 1–5.

[69] Ji S, Li G, Li C, Feng J. Efficient Interactive Fuzzy Keyword Search. In: Proceedings of the 18th International Conference on World Wide Web. WWW '09. New York, NY, USA: ACM; 2009. p. 371–380. Available from: http://doi.acm.org/10.1145/1526709.1526760.

[70] Solutions VI. Approximate Matching (whitepaper); 2008. Accessed 25.04.17. Available from: http://viewds.com/images/pdf/Whitepapers/approximate%20matching%202.pdf.

[71] Elmes GA, Roedl G, Conley J. Forensic GIS: The Role of Geospatial Technologies for Investigating Crime and Providing Evidence. Springer Publishing Company, Incorporated; 2014.

[72] Selvi RT, Raj EGDP. An Approach to Improve Precision and Recall for Ad-hoc Information Retrieval Using SBIR Algorithm. In: 2014 World Congress on Computing and Communication Technologies; 2014. p. 137–141.

[73] Cai J, Shao X, Ma W. Ontology Driven Semantic Search over Structure P2P Network. In: 2009 Ninth International Conference on Hybrid Intelligent Systems. vol. 3; 2009. p. 29–34.

[74] Bonino D, Corno F, Farinetti L, Bosca A. Ontology Driven Semantic Search. In: WSEAS International Journal Multimedia and Image Processing (IJMIP), Volume 2, Issues 1/2, March/June 2012 Copyright © 2012, Infonomics Society 148 Transaction on Information Science and Application, Issue 6; 2004. p. 1597–1605.

[75] Kleppmann M. real time full text search with luwak and samza; 2015. Accessed 29.04.17. Available from: https://www.confluent.io/blog/real-time-full-text-search-with-luwak-and-samza/.

[76] Data I. Building a Streaming Search Platform; 2016. Accessed 29.04.17. Avail-

able from: https://blog.insightdatascience.com/building-a-streaming-search-platform-61a0d5a323a8.

[77] Lillis D, Scanlon M. In: Park JJJH, Jin H, Jeong YS, Khan MK, editors. On the Benefits of Information Retrieval and Information Extraction Techniques Applied to Digital Forensics. Singapore: Springer Singapore; 2016. p. 641–647. Available from: http://dx.doi.org/10.1007/978-981-10-1536-6_83.

[78] Marijan R, Leskovar R. A library's information retrieval system (In)effectiveness: case study. Library Hi Tech. 2015;33(3):369–386.