

Relevance Feedback Optimization for Digital Forensic Investigations

تحسين موائمة الردود في تحقيقات الأدلة الإلكترونية

by

HANADI AL SUWAIDI

**A thesis submitted in fulfilment
of the requirements for the degree of
DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE**

at

The British University in Dubai

February 2019

DECLARATION

I warrant that the content of this research is the direct result of my own work and that any use made in it of published or unpublished copyright material falls within the limits permitted by international copyright conventions.

I understand that a copy of my research will be deposited in the University Library for permanent retention.

I hereby agree that the material mentioned above for which I am author and copyright holder may be copied and distributed by The British University in Dubai for the purposes of research, private study or education and that The British University in Dubai may recover from purchasers the costs incurred in such copying and distribution, where appropriate.

I understand that The British University in Dubai may make a digital copy available in the institutional repository.

I understand that I may apply to the University to retain the right to withhold or to restrict access to my thesis for a period which shall not normally exceed four calendar years from the congregation at which the degree is conferred, the length of the period to be specified in the application, together with the precise reasons for making that application.



Signature of the student

COPYRIGHT AND INFORMATION TO USERS

The author whose copyright is declared on the title page of the work has granted to the British University in Dubai the right to lend his/her research work to users of its library and to make partial or single copies for educational and research use.

The author has also granted permission to the University to keep or make a digital copy for similar use and for the purpose of preservation of the work digitally.

Multiple copying of this work for scholarly purposes may be granted by either the author, the Registrar or the Dean of Education only.

Copying for financial gain shall only be allowed with the author's express permission. Any use of this work in whole or in part shall respect the moral rights of the author to be acknowledged and to reflect in good faith and without detriment the meaning of the content, and the original authorship.

Abstract

Digital forensics deals with the use of tools and techniques to preserve, identify, extract, document, and interpret any data stored or transmitted using a digital system. It is usually used to help support or refute a theory, for the occurrence of an offense or crime, or it might indicate intent or alibi.

There are many challenges when it comes to the forensics discipline of digital evidence, and the sheer amount of data found on modern digital devices is one of them. In today's society, it became the norm for one individual to own multiple digital devices with large storage capacities. If that individual was part of a group of people accused of a certain crime, the end result would be a large amount of data, possibly in Terabytes. Furthermore, such data would usually need to be investigated for evidence in a limited window of time. Digital forensic laboratories that rely on traditional forensic tools usually lack the resources required to handle the size of data found on digital devices today.

The work presented in this thesis can be seen as a step forward into enhancing digital forensics investigations by optimizing the investigator's relevancy feedback. The study proposes a framework that integrates different text processing and mining techniques to assist the examiner reach useful information faster. The framework has been implemented and evaluated using a real world crime dataset of Arabic text. A Proof-of-Concept implementation was evaluated by experienced senior digital forensics examiners. The results showed a good improvement in the average recall-precision rates and a reduction of the required time to complete the tasks by 53% over the time spent using traditional tools.

ملخص

يتضمن فحص الأدلة الإلكترونية استخدام الأدوات والتقنيات للحفاظ على الدليل وتحديد واستخراج وتوثيق وتفسير أي بيانات مخزنة أو منقولة باستخدام نظام إلكتروني. يتم استخدام عمليات الفحص للمساعدة في دعم أو نقض نظرية حدوث جريمة أو جنحه، أو تستخدم للتأكد من وجود دلائل على نية لارتكابها أو ذريعة تنقضها. هناك العديد من التحديات التي تواجه تخصص الأدلة الإلكترونية، منها الحجم الهائل للبيانات التي يتم العثور عليها في الأجهزة الإلكترونية. حيث أصبح من المعتاد في هذا الزمن امتلاك الشخص لعدد من الأجهزة الإلكترونية ذات سعات التخزين الكبيرة.

فإذا كان هذا الشخص جزء من جماعة من الناس المتهمين بجريمة معينة، ستكون نهاية المطاف كم كبير من البيانات، التي قد تحسب بالثوابيت. إضافة لذلك، سيتطلب فحص هذه البيانات للبحث عن أدلة في وقت قصير من الزمن. تعتمد مختبرات فحص الأدلة الإلكترونية على الأدوات التقليدية للفحص وعادة تفترق إلى المصادر المطلوبة للتعامل مع هذا الحجم من البيانات التي يتم العثور عليها في الأجهزة الإلكترونية.

يمكن اعتبار العمل المقدم في هذه الأطروحة خطوة إيجابية لتطوير عمليات التحقيق في الأدلة الإلكترونية من خلال موائمة الردود لفاحص الأدلة الإلكترونية. تقترح الدراسة هيكل تندمج فيه عمليات معالجة واستخراج البيانات لمساعدة الفاحص للوصول بسرعة إلى المعلومات المطلوبة. تم تنفيذ وتقييم الهيكل باستخدام مجموعة بيانات باللغة العربية من جرائم واقعية، وشارك في التقييم فاحصو أدلة إلكترونية متمرسون. أظهرت النتائج تحسين لنتائج الاسترداد والدقة، وتقليص الوقت المستغرق لإكمال العمل المطلوب بنسبة 53% بشكل أسرع من الوقت الذي استغرقه الفاحصون لإنجاز المطلوب باستخدام الأدوات التقليدية.

Dedication

I would like to first send massive appreciation to my Father, Uncle, and Brother for always pushing me forward and supporting me through my ups and downs. Thanks to everyone in my family and my friends for their patience and understanding.

I would like to dedicate this research to my best friend, Hend Al Nuaimi. Thank you for always being there, thank you for constantly cheering me up, and thank you for being extra understanding during the final months of this long voyage.

Acknowledgments

I would like to first thank my advisor, Professor Sherief Abdalla (BUiD) for his counsel and for shining a guiding light on this journey.

I would also like to thank the Emirates Electronic Evidence Center (EEEC) at Abu Dhabi Police, who provided me with great, supportive, and friendly experts to complete this research. My gratitude also goes to all the participants who contributed time and effort to this work.

Finally, I would send thanks to my friend and first mentor in the field of digital forensics, Mr. Ivor Rankin, for his helpful anecdotes and for always being there to discuss problems and provide support.

Table of Contents

Chapter One: Introduction.....	1
1.1. Introduction to Text Data Mining	4
1.2. Introduction to Relevance Feedback	5
1.3. Research Motivation.....	7
1.4. Outcome	11
1.5. Research Question & Aims.....	12
1.6. Contributions	15
1.7. Scope of Dissertation	17
Chapter Two: Literature Review	19
2.1. Digital Forensics	19
2.1.1. Data Reduction & Triage	19
2.1.2. Text Data Mining (TDM).....	22
2.1.3. User Profiling & Other Solutions.....	30
2.1.4. Summary of Digital Forensics Review	35
2.2. Text Data Mining (TDM).....	42
2.2.1. Clustering	42
2.2.2. Classification	47
2.2.3. Summary of TDM Review.....	52
2.3. Relevance Feedback.....	53
Chapter Three: Text Data Mining Background	63
3.1. Clustering	64
3.1.1. K-Means Clustering	66
3.1.2. Hierarchical Clustering	68
3.2. Classification	70
Chapter Four: Research Approach.....	73
4.1. Research Design	73
4.2. Reliability.....	76
4.3. Ethical Considerations.....	77
Chapter Five: Proposed Framework	79

5.1.	Framework Components	79
5.2.	Framework Workflow	82
5.3.	Search Concepts.....	83
5.4.	Proof-of-Concept Preview.....	85
Chapter Six: Experiments Design		92
6.1.	Experiments Goals	92
6.2.	Experiments Dataset.....	93
6.3.	Evaluation Measurements	97
Chapter Seven: Proposed Framework Components Testing		100
7.1.	Clustering Component Experiment.....	100
7.2.	Classification Component Experiment.....	102
7.3.	Search & Visualization Experiment	105
7.3.1.	Keyword Search Test.....	106
7.3.2.	Visualization Assisted Search Test	107
7.4.	PoC Experiment.....	112
Chapter Eight: Evaluation of Results.....		115
8.1.	Clustering Component Experiment Results	115
8.2.	Classification Component Experiment Results	115
8.3.	Search & Visualization Experiment Results.....	117
8.4.	PoC Experiment Results	118
8.5.	Interviews & Questionnaire Results	120
Chapter Nine: Discussion		123
9.1.	Discussion of Experiments	123
9.2.	Discussion of Interviews.....	126
9.3.	Key Findings	128
9.4.	Observations	129
Chapter Ten: Conclusion & Recommendations		132
10.1.	Conclusion.....	132
10.2.	Limitations.....	134
10.3.	Recommendations	137
10.4.	Future Work.....	138
References		140

Appendix	154
1. Appendix – Research Authorization Letter	154
2. Appendix – Search & Visualization Experiment	155
2.1. Setup with dtSearch (Test 1)	155
2.2. Setup with RapidMiner (Test 2)	157
2.3. Test Questions	163
2.4. File Clusters & Word Clouds	164
3. Appendix – PoC Filters & Operators	173
3.1. RapidMiner Filters (Preprocessing)	173
3.2. RapidMiner Operators	174
3.2.1. Step 1 (Clustering)	174
3.2.2. Step 2 (Classification)	176
3.2.3. Step 3 (Apply Model)	179
3.2.4. Other Operators	180
4. Appendix – Detailed Research Results	184
4.1. Detailed PoC Experiment Results	184
4.1.1. Participant 1 Results	184
4.1.2. Participant 2 Results	186
4.1.3. Participant 3 Results	187
4.1.4. Participants Combined Result Averages	190
4.1.5. Percentage of Change (Time Analysis)	191
4.2. Interview & Survey Results	192
4.2.1. Interview Combined Transcripts (Pre–Tool Testing)	192
4.2.2. Survey (Post–Tool Testing)	198
4.2.3. Questionnaire Results	199

List of Figures

Figure 3.1: Unsupervised Algorithms (Baeza-Yates & Riberio-Neto, 2011, p. 284).....	64
Figure 3.2: Supervised Algorithms (Baeza-Yates & Riberio-Neto, 2011, p. 285)	65
Figure 5.1: Proposed Framework.....	80
Figure 5.2: Options Menu.....	86
Figure 5.3: Term Search Wizard.....	86
Figure 5.4: Term Search (Review)	87
Figure 5.5: Term Search (View File).....	87
Figure 5.6: Term Search (Word Cloud).....	87
Figure 5.7: Model & Classify	88
Figure 5.8: Cluster Model.....	88
Figure 5.9: Classification Results (Review).....	89
Figure 5.10: Classification Results (Options)	89
Figure 5.11: The More Like This Option	90
Figure 5.12: Classify by Relevant DB.....	90
Figure 5.13: View RelevantDB	91
Figure 7.1: Document Clustering Process	102
Figure 7.2: Document Processing with k-NN	103
Figure 7.3: Document Processing with kNN.....	104
Figure 7.4: Applying the Learnt Model on the Test Set.....	104
Figure 7.5: Screenshots of Test Website Pages (Main Page)	108
Figure 7.6: Screenshot of a Test Website Page.....	109
Figure 8.1: dtSearch Processed Documents Report	117
Figure 8.2: RapidMiner Execution Time (Main & Sub Clusters)	117
Figure 9.1: Experiments Accuracy Results (Encase vs. Probo)	125
Figure 1.1: Research Authorization Letter.....	154
Figure 2.1: dtSearch Desktop Search Feature	155
Figure 2.2: dtSearch Desktop Search History (Test 1).....	156
Figure 2.3: Processing Documents Operators	157
Figure 2.4: Clustering of Processed Documents.....	158
Figure 2.5: Main Page to Navigate to Groups (Main Clusters).....	160
Figure 2.6: Each Group (Main Cluster).....	161
Figure 2.7: Subgroups (Sub Clusters).....	162
Figure 3.1: Document Clustering Process in RapidMiner Studio.....	174
Figure 4.1: Filled Up Survey (Participant 1)	200
Figure 4.2: Filled Up Survey (Participant 2)	200
Figure 4.3: Filled Up Survey (Participant 3)	200
Figure 4.4: Filled Up Questionnaire (Participant 1).....	200
Figure 4.5: Filled Up Questionnaire (Participant 2).....	200
Figure 4.6: Filled Up Questionnaire (Participant 3).....	200

List of Tables

Table 1.1: FBI RFCL Annual Reports 2003-2016	8
Table 3.1: Accuracy Estimation	72
Table 5.1: Proposed System Components	81
Table 5.2: User Interface Main Sections	82
Table 6.1: Experiments Goals & Relation to Research Aims	92
Table 6.2: Experiments & Datasets	94
Table 6.3: Training & Test Sets File Distribution	96
Table 7.1: Cluster Category	108
Table 7.2: Main Clusters Word Clouds & Sub-Clusters Files	111
Table 7.3: Groups Distribution over Tests	113
Table 8.1: Clustering Comparison Results	115
Table 8.2: Classification Algorithms Results – Training Stage (Using Training Set)	116
Table 8.3: Classification Algorithms Results – Test Stage (Using Test Set)	116
Table 8.4: Classification Algorithms Results	116
Table 8.5: Experiment Timing	118
Table 8.6: Experiment Results for Each Participant (Encase vs. Probo)	119
Table 8.7: Experiment Results for Each Tool (Combined Total % Avg.)	120
Table 9.1: Experiment Result Total % Averages (Participants Combined Avg.)	126
Table 2.1: Top 20 Terms from Main Clusters	164
Table 2.2: Top 10 Terms from Sub Clusters of (Main Cluster 0)	165
Table 2.3: Top 10 Terms from Sub Clusters of (Main Cluster 1)	165
Table 2.4: Top 10 Terms from Sub Clusters of (Main Cluster 2)	165
Table 2.5: Top 10 Terms from Sub Clusters of (Main Cluster 3)	166
Table 2.6: Top 10 Terms from Sub Clusters of (Main Cluster 4)	166
Table 2.7: Top 10 Terms from Sub Clusters of (Main Cluster 5)	166
Table 2.8: Sub Clusters’ Files & Word Clouds	172
Table 4.1: Participant 1 Experiment Results (Encase)	185
Table 4.2: Participant 1 Experiment Results (Probo)	186
Table 4.3: Participant 2 Experiment Results (Encase)	186
Table 4.4: Participant 2 Experiment Results (Probo)	187
Table 4.5: Participant 3 Experiment Results (Encase)	188
Table 4.6: Participant 3 Experiment Results (Probo)	189
Table 4.7: Participants Combined Result Averages	190
Table 4.8: Percentage of Change (Time Analysis)	191
Table 4.9: Combined Participants Questionnaire Results (Post-Experiments)	200

Chapter One: Introduction

The rapid technological advancements in the 21st century caused the field of digital forensics to be in a constant state of change, as modern digital devices became ubiquitous. Consequently, there are many challenges when it comes to the Digital Forensics (DF) field, which led to the continuous improvement in procedures, tools, and methods.

This research started out from the simple idea that current DF tools lack the power that could be gained from relying on the relevancy feedback of DF investigators during their analysis of cases. The main challenge the study focused on was the high information retrieval overhead problem.

In other words, the high rates of false positive and low rates of true positive in the results returned by current tools. This problem often leads to an investigator's time being spent reviewing thousands of search hits that are non-relevant. Therefore, the current work proposed a framework that integrates text data mining techniques and an investigator's relevancy feedback to speed up the process of finding useful information.

Corporate and law enforcement agencies worldwide use DF investigations to help them resolve traditional crimes, incidents, or cyber crimes by analyzing data from related digital devices.

Contrary to what many believe the background of how DF matured over the past decade; a forensic laboratory was not where this discipline began.

Instead, police officers and detectives who had some expertise or interest in computers viewed them as adding evidentiary value to their investigations. Hence, over the years this discipline underwent rapid development as it became more routine and under scrutiny from the other fields of the forensic science. (Council, 2009, p. 181)

DF investigators have to deal with the ever expanding storage space of digital devices and the increasing numbers of devices per individual in society; as those devices keep getting cheaper. For instance, large-scale criminal investigations might involve the examination of tens or hundreds of devices, which makes it infeasible within a short amount of time and limited human resources. Therefore, the DF research communities continue in their efforts to develop better methods as new challenges arise.

One definition of Digital Forensics is the “analytical and investigative tools and techniques for the preservation, identification, extraction, documentation, analysis and interpretation of digital media ... and the presentation of digital evidence.” (Grobler, 2011)

In order to better understand the main goal of this study, it is critical to understand the meaning of a DF investigation. The title “Digital Forensics Investigation” has three main components; namely Digital Evidence, Digital Forensics, and Digital Investigation.

ACPO (2005) defined digital evidence as “information and data of investigative value that is stored on or transmitted by a computer”. Casey (2011) added another definition for digital evidence as “any data stored or transmitted using a computer that support or refute a theory of how an offense occurred or that address critical elements of the offense such as intent or alibi”.

On the other hand, as explained by Kruse & Heiser (2001), digital forensics deals with the “preservation, identification, extraction, documentation and interpretation of computer data”.

From a law enforcement perspective, a digital device can be found in a case in two main situations.

First, if it was used to commit a crime or if it was a target of a crime (i.e. victim's device) then a case becomes purely a cyber crime related investigation. Second, if digital devices were present in a crime scene or belonged to person(s) of interest in a case, then it might contain data of evidentiary value to an investigation.

An investigation is a systematic examination of something. It is regrettable that sometimes digital forensics is misunderstood as being different from other types of investigations.

For example, during a murder investigation that took place in a living room, the crime scene would be photographed, the investigative team would search for evidence, and take samples from the scene. Similarly, during a digital forensics investigation, evidence collection would proceed in a similar fashion. However, sometimes people expect an entire system to be recreated during a digital forensics investigation, which is not the case in most instances. Contrary to what is shown on some TV crime procedural dramas. (Casey, 2010)

Digital forensics investigations usually follow certain workflows based on the Standard Operating Procedures (SOP) of an organization or laboratory. The flow of digital evidence should be traceable from the incident or crime scene, where it was first found and identified, to how it would be presented in court or to higher management.

Pollitt (1995) introduced a four step digital forensics model to ensure the admissibility of digital evidence in a court of law. The four steps were acquisition, identification, evaluation, and admission of evidence. In 2001, the Digital Forensics Research Workshop (DFRWS) proposed the digital investigative model that contained six stages. These stages were identification, preservation, collection, examination, analysis, and presentation. (Johansen, 2017, p. 33)

Researchers and tool developers are focused on helping the digital forensics investigator find results faster, through automation of data preprocessing prior to the beginning of the analysis phase. Furthermore, in previous research Text Data Mining techniques (Beebe and Clark 2007; Albalate, et al. 2010) and Relevance Feedback (Beebe, et al. 2011; Varma, et al. 2014) had been associated individually with digital forensics.

In contrast, this work proposes a framework that uses Text Data Mining (TDM) and Relevance Feedback (RF) techniques to improve the accuracy of retrieved information, which would be presented to the digital forensics investigator during the case analysis phase. The following sections provide introductions to TDM and RF, the research motivation, expected outcome, research question and aims, contributions, and the scope of this dissertation.

1.1. Introduction to Text Data Mining

Text Data Mining (TDM), according to Nemati & Barko (2003), is “the science and art of extracting meaningful factual information from masses of text, usually by means other than the statistical approaches that have produced value in numerical and fixed-field data”.

Text Data Mining, also known as Text Mining, differs from Information Retrieval (IR). Hearst (1999) pointed out that the goal of IR was to help a user in finding documents with information that satisfy their needs, but the selection of the required information is left to the user.

Text Mining differs from Data Mining (DM). Text mining, as discussed by Makhabel, et al. (2017), is concerned with the extraction of relevant information from large text of natural language and the search for interesting relationships or semantic associations. In contrast, the goal of DM is to help the user in the discovery of new information from data by separating signal from noise or to

help them recognize patterns across datasets. (Hearst, 1999)

In contrast to DM and IR, Text Mining deals with retrieval of information from documents as well as attempts to find new patterns of information, which could be useful, non-trivial, and unknown previously. (Alwidian, et al., 2015)

A fully featured solution should combine the best of both TDM and IR capabilities. Liddy (2000) asserted that such a technology would be able to identify sources worth mining that contain relevant information, extract entities with valuable knowledge, generate and efficiently store semantic interpretations of the information, and would be able to offer ways that such information would be easily accessed and utilized.

In both TDM and DM, a common requirement is that extracted information ought to be potentially useful. The main problem with text is that it could be unstructured or difficult to deal with, contrary to the data dealt with by DM techniques. The purpose of TDM is to help in the analysis and extraction of useful information for particular reasons. (Witten, et al., 2016, p. 515)

1.2. Introduction to Relevance Feedback

Turnbull & Berryman (2016, p. 2) defined Relevance as “the art of ranking content for a search based on how much that content satisfies the needs of the user and the business”. Relevance Feedback (RF) can be defined as a “technique of marking one or more results, either manually or automatically, as relevant, and then using the important terms to form a new query.” (Ingersoll, et al., 2013, p. 78)

Relevance Feedback, as explained by Rocchio (1965), is when a process is developed based on a sequence of retrieval operations in order to modify a request. A user would communicate to the

system their evaluation after each operation, which will be the basis of altering that user's query.

Search engines typically depend on a user's query in order to provide them with information. Usually a short query of terms is used by the user to search for specific information that relates to their topic of interest, which is challenging. As generally little context is found within those terms. Relevance feedback techniques were proposed to help in the incorporation of the subjectivity of the human perception in the retrieval process, by giving the users the chance to evaluate retrieved results. (Guan, et al., 2017, p. 318)

Additionally, one of the major problems with generic search engines over the Internet is that they do not use knowledge about the user or the data source they cover. In the context of a digital forensics investigation, the underlying information about a case can be used when a search query is used to retrieve relevant information from a document collection.

Contrary to the seemingly infinite number of documents one envisions while running a query using an online search engine, i.e. Google, there is a finite number of documents one encounters during a digital forensics examination of a device. Thus, a methodical search of the data is usually conducted to speed up the data analysis process by the digital forensics investigator to help them locate relevant information faster.

Digital forensic investigators encounter an increasing amount of digital evidence related to the suspects, victims, and possible criminal activities. This evidence could be hidden within a significant amount of digital data generated by everyday activities.

1.3. Research Motivation

One of the main challenges that face Digital Forensics (DF) investigators is the sheer amount of data found on digital devices. The amount of digital data will only be expanding in the future. According to Reinsel, et al. (2018), the International Data Corporation estimated that on a daily basis more than 5 billion consumers interact with data. That number is expected to increase by 2025 to 6 billion, meaning 75% of the world's population.

In 2017, the individual cybercrime victim lost an average of \$142, globally. On the other hand, consumers who were cybercrime victims in the United Arab Emirates (UAE) lost a total of \$1.1 Billion. And while 24% of global cyber incidents occurred in Europe and the Middle East & Africa (EMEA) region, North America was the most targeted in 2018 with up to 43% of global cyber incidents. (Sharma, 2018)

The FBI's annual reports shared that between the years 2003 and 2016, the average case size in 2003 was 83 Gigabytes (GB), and by 2016 it grew to reach 1,084 GB of data, as shown in (Table 1.1). Furthermore, the total volume of data increased by an average of 67% from 82 Terabytes (TB) in 2003 to 5,986 TB in 2012. (Quick & Choo, 2018, p. 13)

Not only is the volume of data increasing, but also the number of devices. In a study by Facebook-IQ (2016) it was found that 98% of teens in Germany and 94% of teens in France own multiple devices.

In the UAE, Alawadhi, et al. (2015) used real cases data records and statistics from a 12 years' period, to analyze the different factors that influenced the investigations done by Dubai Police's Digital Forensics Department. The study estimated an increase of 20% in the average volume of

evidence items per case between the years 2011 and 2014. The average per case in 2010 of 171 GB increased to 900 GB then to 1,186 GB in 2011 and 2014, respectively.

US fiscal year	Service requests received	Examinations conducted	TB processed	Average case size (GB)
2003	1,444	987	82	83
2004	1,548	1,304	229	176
2005	3,434	2,977	457	154
2006	4,214	3,633	916	252
2007	4,567	4,634	1,288	278
2008	5,057	4,524	1,756	388
2009	5,616	6,016	2,334	388
2010	5,985	6,564	3,086	470
2011	6,318	7,629	4,263	559
2012	5,060	8,566	5,986	699
2013	6,040	7,273	5,973	821
2014	6,994	6,322	5,060	800
2015	6,321	5,897	5,276	895
2016	5,939	5,229	5,667	1084
Total	68,537	71,555	42,373	

Furthermore, the number of cases increased on a yearly basis throughout the study sample. In 2003 there were 51 cases, and by the year 2013 there were more than 900 cases. Additionally, they found that there was an increase over the 12 years in the number of cases that took more than 200 hours to examine, which peaked at 2011. For example, it took 4,368 hours to investigate a case with 64 evidence items with a volume size of 28 TB. (Alawadhi, et al., 2015)

In general, the time used to resolve a case depends on many elements, including the media size, case complexity, and the amount of information that needs to be reviewed. The current research formulated a new framework that could help yield faster results during digital forensics examinations.

A digital forensics investigator in a typical case could weed out routine jobs, use file signatures or

hashing to remove known files, or use keyword searches to reduce the amount of data to be analyzed. Nonetheless, they would still be required to shuffle through thousands of files, all while working against the clock. Today, even with all the developed digital forensics techniques, there is still something missing for this investigative formula to be more effective.

Furthermore, the evaluation of newly developed techniques in the digital forensics field can be limited because of the difficulty in testing most of those techniques on real world criminal datasets, which Fahdi, et al. (2016) stressed on in their study. Furthermore, Grajeda, Breitingner, and Baggili (2017) discussed and detailed the lack of dataset sources as being one of the major challenges in cyber forensics.

There are many techniques that are helpful during a digital forensics examination. The use of case related information to do a String Search, better known as a “Keyword Search” is of interest and relation to this study. A list that consists of related terms is usually generated at the beginning or during the analysis phase of the investigation, in order to find more related information. There are many limitations when it comes to this technique; of significance are high recall and low precision rates.

This study aimed to develop an investigator’s approach to tackling the problem of the high information retrieval overhead, which is normally generated by digital forensics tools. As the traditional search in digital forensics seeks recall rates at or near 100%. However, the query hits’ precision rate is usually low.

In time-sensitive cases, an investigator views recovered evidence as valuable intelligence that would help them swiftly carry out the remainder of their investigation. For example, in child

exploitation, abuse, or kidnapping cases time is often of the essence. Therefore, an investigator with the correct information might be able to protect current and possible future victims.

Most digital forensics tools that employ the Keyword Search feature present all matching results to the investigator, who then sifts through them to reach relevant information, if any. For instance, a keyword search for a term like 'kill'¹ in a homicide case might result in million hits, as the forensic tool searches the media under examination regardless of the context of where such a term occurs. In this case, the human analytical time for the examiner would be spent analyzing those hits. (Beebe & Dietrich, 2007)

Alzaabi, et al. (2013) argued that the decreasing costs of devices along with the increase of their storage capacity constitute a serious challenge in terms of time and effort for the investigator, even with the availability of tools and techniques to assist them. The large number of files that is required to be processed during a digital forensics investigation is one of its biggest challenges.

The use of reference databases was a viable solution to this problem, when the number of files was manageable, but with the increase in both storage capacity and device numbers in an investigation it became obsolete. As filtering of known files or file hashing becomes useless when the underlying data is regularly altered. (Breitinger & Rousev, 2014)

Recall plays an important factor to many search tools, and maintaining recall while increasing precision is favorable when it comes to the digital forensics domain. Beebe (2009) argued that smarter analytical algorithms would enable examiners reach relevant data faster, help reduce the

¹ Note here that such a literal string search would result in many false positives; as the tool would also retrieve system commands and any other occurrence of this term, regardless of its relevance.

noise filtered out by the examiners, and help in the conversion of data into useful information and knowledge.

The audit report by the Department of Justice (2016) in the United States of America (USA), explained that larger disk storage in everyday digital devices increases the time needed to complete digital forensics investigations. This fact was contributing to the growing case backlogs at digital forensics laboratories.

There are serious implications to the increasing number of backlogged cases. The legal maxim of “justice delayed is justice denied” is at the forefront of these implications. A delay in case proceedings, because of the time waiting for the results of the digital forensics analysis, might result in reduced sentences for convicted defendants. Moreover, while waiting for the analysis to finish, a suspect might be denied access to their family. (Shaw & Browne, 2013)

Additionally, in lengthy trials a suspect would have general difficulties in life, as a consequence of their work and livelihoods being affected, even if they were found innocent at the end. Gogolin (2010) surveyed law enforcement agencies in Michigan in the USA, and reported that a digital component was in 50% of cases. In addition, digital forensics labs reported a backlog of two years.

1.4. Outcome

The proposed framework would provide the digital forensics investigator with a better Information Retrieval solution to combat the high overhead of search hits generated by current forensic tools. Additionally, this research highlights the effectiveness of Text Data Mining (TDM) and Relevance Feedback (RF) techniques in light of digital forensics investigations.

The advantages added by the proposed framework seek to cut down the time for case turn-over, through applied intelligence and user feedback. It would also help improve the efficiency and accuracy of findings, while preserving evidence integrity. This would be achieved by using TDM techniques and optimizing the relevancy feedback from the digital forensics investigator on processed results.

1.5. Research Question & Aims

This study endeavored to answer the following question:

How can the Digital Forensic discipline utilize Text Data Mining and Relevance Feedback techniques to achieve optimal results faster?

There were three main research aims that directly map from the main research question. They include the following:

1. Establish the need for the use of Text Data Mining (TDM) and Relevance Feedback (RF) techniques in Digital Forensics (DF).
2. Propose a framework to incorporate TDM and an investigator's relevancy feedback.
3. Implement a Proof-of-Concept (PoC) for the proposed framework and evaluate its results.

There is a lack of appreciation for the discipline of relevance feedback when it comes to digital forensics investigations, not through a deliberate disregard for its benefits, but this stems from a lack of real understanding of the obvious benefits it can provide. Although, recent studies like Beebe, et al. (2011) and Varma, et al. (2014) took note of these benefits and aimed their research to highlight the adaptation of both TDM and RF techniques, respectively, to the field of digital forensics.

DF investigations are useful in both conventional and cyber crimes. The difference between these two types is that cybercrime can be defined as any unlawful activity that uses a digital system as a tool, target, or a means for carrying out further crimes. A conventional crime or offence is a “legal wrong that can be followed by criminal proceedings which may result in punishment”. (Hudson, 1994, p. 27)

DF investigations help in a variety of cases; for example civil cases, criminal cases, Human Resource cases, corporate espionage, incident response, risk assessment, etc. In order for a forensic examiner to fulfill their duty in providing accurate evidence that relates to an investigation in an unbiased and objective way; the scientific method is one of their biggest assets.

The scientific method, as explained by Casey (2010, p. 6), typically starts with the gathering of facts, and based on the available evidence a hypothesis is formed. It is important to keep in mind that some observations or analysis could be incorrect. The veracity of a hypothesis can be assessed by looking for supporting evidence and the mental flexibility to consider other possibilities.

Digital forensics labs are overwhelmed by the sheer number of data they have to handle per case, which causes an increase in the number of backlogged cases, as confirmed by Gogolin (2010) and the Department of Justice (2016). Instead of overworking to capacity the limited resources most laboratories have to handle their backlogged cases, it is hardly an argument that the smart thing to do would be to direct current research to Artificial Intelligence and Data Mining to lower that load.

There are many challenges when it comes to the field of DF. One of the fundamental issues is that the tools were originally designed to help find evidence where the possession of that evidence in itself was the crime, i.e. child pornography.

As explained by Garfinkel (2010), the current tools lack the capability of identifying out-of-the-ordinary or subtly modified information. And while those tools are able to work with several Terabytes worth of data, they cannot practically assimilate them into a coherent report. Additionally, it is not easy to use these tools for the reconstruction of a perpetrator's actions or past events, as it is more or less left to be done manually by the examiner.

The same conceptual model was implemented by most of the current digital evidence analysis tools, which Garfinkel termed the "Visibility, Filter and Report" model. In this model, all the collected data would be presented (Visible) to the examiner, who then Filters the data to reach relevant information, and finally a Report would be generated of what was found. (Garfinkel, 2010)

The use of intelligent analytical algorithms in the context of digital forensics was discussed by Beebe (2009), who pointed out that the current approaches of search, retrieval, and analysis of digital evidence relied largely on:

- i.** Literal String searching; i.e. non-GREP string searches for text and file signatures.
- ii.** Simple Pattern matching; i.e. GREP searches.
- iii.** Indexing data to speed up searching and matching.
- iv.** Hash Analyses.
- v.** Logical level file reviews; i.e. log analysis, registry analysis, Internet browser file parsing,

etc.

Additionally, Garfinkel (2010) argued that in order to move forward the community must adopt modular approaches for the forensic processing and data representation. Garfinkel also highlighted some of the major issues facing digital forensics research. He concluded that the research community can improve the quality of its research efforts and lower the development costs at the same time; by paying careful attention to cooperation, and shared development of standardized ways of thinking about, representing and computing data.

On the other hand, Wiles (2011, p. 144) suggested that over the next ten years the specialization of labor will be defining the forensic and e-discovery space, and that it will be hard for a solo examiner to handle certain investigations because of their scale. Moreover, more bodies will be required to conduct such investigations, because the amount of data that needs to be waded through is getting too large.

It is difficult to come up with one solution that could fit all investigations. Casey (2010, p. 13) attributed that to the difficulty of creating Standard Operating Procedures (SOP) for all in-depth forensic analysis, as every investigation is different. Therefore, having a methodical approach for the organization and analysis of large amounts of data is important.

DF investigators are always looking for faster methods to recover constructive information with evidentiary value from digital media. Thus, finding ways on how to provide the evidence quickly, while preserving its integrity, is a challenge.

1.6. Contributions

The main contributions of this work are as follows:

1. The study established that there was a lack of a practical file's grouping technique in traditional digital forensic tools and that relevant information found during the analysis can be used in a limited way and only manually by the investigator. This was established as follows:
 - a. Semi-structured interviews were conducted with the study participants.
 - b. The responses showed a lack of similarity techniques to automate the process for the grouping of useful information. To put it another way, short of finding case-specific keywords or exact duplicates of files through MD5 hashing, a digital forensic investigator would be left to manually examine thousands of files.
2. A new framework was proposed to incorporate TDM and an investigator's relevancy feedback into the digital forensic investigation. The development of the framework relied on the following:
 - a. In order to find out the best algorithms that could be integrated into the proposed solution, a survey for the state of art on the best TDM and RF techniques was done.
 - b. The selection of the best algorithms to use for each component and visualization of results were based on experiments. This resulted in the selection of the K-Means and kNN for the clustering and classification algorithms, respectively.
 - c. The selection of the best visualization option to help the user assimilate the information given by the TDM components, which was clustering with word clouds.

3. A Proof-of-Concept (PoC) implementation was built for the proposed framework and tested by the participants against traditional DF tools. The evaluation of the PoC relied on an experiment, a survey, and a questionnaire that were conducted as follows:
 - a. The researcher tested the PoC during the training phase.
 - b. The study participants tested the PoC during the test phase and completed the surveys and questionnaires after the test.
 - c. The evaluation of the results relied on the selected evaluation measures.

1.7. Scope of Dissertation

The outline of this study is as follows:

- **Chapter One:** the current chapter provides an introduction to the problem and explores the research motivation, question, aims, and contributions.
- **Chapter Two:** this chapter explores the literature and what had been done previously in the main areas of interest, including Digital Forensics, Text Data Mining, and Relevance Feedback.
- **Chapter Three:** This chapter explores the concepts of the clustering and classification components of the proposed framework.
- **Chapter Four:** this chapter explains the research approach and justification. It also discusses the research reliability and ethical considerations.
- **Chapter Five:** this chapter shares information on the proposed framework, its components and workflow, and previews the developed Proof-of-Concept (PoC) tool.
- **Chapter Six:** this chapter explains the setup for all the experiments along with the used datasets composition and evaluation measures.

- **Chapter Seven:** this chapter discusses the performed experiments and explains the used testing methods.
- **Chapter Eight:** this chapter presents the results of all the experiments that were outlined in the previous chapter. In addition, it provides interpretations for the interviews and survey results.
- **Chapter Nine:** the purpose of this part of the study is to discuss the results and key findings, and to provide analysis on the experiments and participants interviews. Additionally, it shares the problems and observations that were encountered during the experiments.
- **Chapter Ten:** this chapter concludes the research and shares the study limitations, recommendations, and future work.

Chapter Two: Literature Review

This chapter reviews the published research on Digital Forensics (DF), Text Data Mining (TDM), and Relevance Feedback (RF). A range of academic databases were surveyed to locate related publications to the research domain. The reviewed databases included the ACM Digital Library, ScienceDirect, Google Scholar, and IEEE Xplore.

2.1. Digital Forensics

The high information retrieval overhead problem and the increase of digital forensics examination times are amplified by the ever expanding volume of disk storage². A variety of research was undertaken in relation to these challenges and different solutions were proposed to handle them. For example, forensic triage, data reduction, data mining, and user profiling.

The need for digital evidence spurred the rapid development of many digital forensic techniques. The following sections present a review of the existing literature on techniques and solutions to the above DF challenges. The last section summarizes the lessons learned from the undertaken review and discusses the gap that was explored by the current work.

2.1.1. Data Reduction & Triage

In 2012, the Dirim tool that was developed by Rowe & Garfinkel (2012) used a system's directory metadata information to automatically find suspicious files in a large corpus. Metadata of a file could include the filename, size, and extension. The tool used drive statistics and compared

² Further details can be found in Section 1.3: Research Motivation.

predefined semantic groups and file clusters. They conducted several experiments on 1,467 drive images corpus with over 8 million files, and found 6,983 suspicious files.

The use of known hash values for data reduction is an old technique in terms of digital forensic practices. Hashing is a useful time saving technique typically used during a DF investigation to filter out known files on the examined device, like the operating system files, or to find specific files of interest.

The hashing method could also be used during forensic triage process to identify relevant devices in a case when there is a known hash or hash list to compare it with. For example, Interpol (2018) maintains a database for child sexual exploitation multimedia hashes³, which could be used to run a search for matching files on suspected devices.

One drawback of relying on the hashing method for DF investigation is the fact that a minor change in the file's data would result in a different hash value. Another downside is that using current processes to compare known hash values is time consuming. (Breitinger & Rousev, 2014)

Rousev & Quates (2012) used similarity digests for content-based forensic triage on the M57 case study which consisted of a 1.5 TB of raw data. There is a difference between the normal cryptographic hashes and similarity schemes. The former seeks to match exact duplicates of the data, while the latter attempts to find objects with non-trivial similarities within their bit stream representation.

³ For instance, Interpol maintain such a database on:
<https://www.interpol.int/en/How-we-work/Databases/International-Child-Sexual-Exploitation-database>

Their study demonstrated that the scope of the investigation could be narrowed down by applying similarity digests in a systematic manner. The typical examination and correlation of a dataset of that size might require a number of days, while their approach was able to examine and triage the dataset in about 40 minutes.

A framework was proposed by Mohammed, et al. (2016) to integrate heterogeneous big data from many sources, and then perform automated forensic analysis on it. The study focused on three main issues, namely data volume, heterogeneous data, and an investigator's problem in understanding the relationship between the many artifacts.

In order for the proposed approach to solve these main issues, it used metadata to solve the first issue and used semantic web ontologies for the second issue. Moreover, to solve the last issue the approach suggested the use of automated identification and correlation of artifacts through artificial intelligence models.

Bharadwaj & Singh (2018) attempted to address the issue of the large volume of data in devices in the DF investigation. They presented a framework that used random sampling method and sector hashing to efficiently investigate traces of target data. The overall idea was to divide the media storage into equal sized regions and select random samples from consecutive regions and compare it to the target data sector. The target data sector hashes would be pre-computed and used to identify if they were found in the media.

The efficiency of their proposed framework was tested using experiments where the target data ranged in size between a few MB to 1 GB, while the disk storage differed in capacities between 4GB, 8GB, 16GB, and 1TB HDD. The reliability of the sampling method relied on two metrics,

namely the number of read requests and IO rate. Based on the results it was observed that regardless of the media storage capacity, the required quantity of random samples would be based on the size of the target data.

A digital forensics data reduction framework was proposed by Quick & Choo (2018) that was built on a common DF framework with added stages; namely data reduction, quick review, and external source data input stages. They started with the use of selective imaging and that only the key files and data would be selected. The decision to include or exclude a particular file type would be left to the forensic investigator. In order to ensure the validity and applicability of the framework, it was applied to a test dataset and a selection of real-world cases, the latter of which was provided by the South Australia Police's Electronic Crime Section.

The experiments' results showed a reduction to 0.206% of the original data, meaning a data source of 8.57 Terabytes of data was reduced to a subset of 12.3 Gigabytes. This subset of data was then explored using quick analysis and semi-automated information and entity extraction.

Additionally, the framework provided a capability to go over the full dataset, if no evidence could be located in the generated subset. Furthermore, the proposed framework could be used to assess which devices might contain valuable evidence and mainly work as a triaging step to prioritize the devices that needed to be focused on. The most common places for potential relevant information could be from found files, such as registry files, Internet Browser History files, Log files, Word Documents, Emails, Spreadsheets, and other system files.

2.1.2. Text Data Mining (TDM)

The application of a Self-Organizing Map (SOM) was discussed by Fei, et al. (2005), to search for

patterns in datasets and visualize similarities in the data. SOM is an unsupervised neural network approach. The study demonstrated the SOM visualization on a dataset of 2,640 graphical images. A noted drawback to this method is that in order to use the data it needs to be transformed manually before it could be used by an SOM application.

In 2007, Khan, et al. (2007) highlighted the issue of the increased data volume, which would result in the need for additional costs and resources. They reasoned that since much of the data is comprised of text then solutions that used text mining methodologies would be needed.

Beebe & Clark (2007) proposed the use of Kohonen's Self-Organizing Maps (SOM) for the post-retrieval clustering of DF text string search results. The main purpose of their research was to test the feasibility of thematically clustering DF text string search results. Their initial pilot tests showed promising results on the clustering quality and the overall utility of the proposed approach. These results were not shared, but according to the researchers it gave them the assurance that scalability would not be an issue based on the observed resource utilization and processing times during the pilot tests.

In 2011, Beebe, et al. (2011) used a Self-Organizing Neural Network (Kohonen Self-Organizing Map) to conceptually cluster the retrieved search hits during a real-world DF investigation and measure IR effectiveness of the new approach by using the precision, recall, and overhead rates. The first and primary goal of their research was to apply text mining techniques to the DF domain in order to reduce IR overhead while achieving at or near 100% recall rates. The secondary purpose was to determine if post-retrieval clustering, as an unsupervised text categorization tool, was capable of increasing the average precision and reducing the IR overhead in general.

They demonstrated improved IR efficiency of the DF text string search while using one real-world case and an experimental one. Based on the empirical results of their research, the clustering process significantly reduced the IR overhead of the DF text string search process.

Additionally, they paid attention in the beginning of their study to different approaches to help them increase average precision, such as relevancy ranking algorithms, visualization, clustering techniques and relevancy feedback systems. Moreover, they reasoned that based on the nature of data in a DF investigation, clustering techniques had a higher probability of success than the others. Because they recognized the inapplicability of most traditional relevancy ranking variables to the DF text string search process.

They also concluded that it was best to start their research by applying clustering only as an automated text categorization tool. This was motivated in part by their desire to measure the impact of clustering as a topical mapping tool on the IR overhead in the context of a DF text string search.

In 2012, Alkoffash (2012) tested the K-Means and K-Medoids algorithms using a manual set of clusters, which consisted of 242 predefined clustered Arabic text documents. The issues facing the K-Means and K-Medoids were represented by problems such as the selection of initial points, and the differing sizes and density of data. Based on the study, the results showed that the average precision and recall were 0.56 and 0.52 for K-Means and were 0.69 and 0.60 for K-Medoids, respectively. These were a good indication for the use of these algorithms, especially for the K-Medoids, which could be applied to Arabic text.

In 2014, Thilagavathi & Anitha (2014) proposed the use of a subject-based semantic Document

Clustering algorithm with a Bisecting K-Means in a hybrid approach, to allow the forensic examiner to cluster documents based on a particular subject. WordNet was used to extract term synonyms and a word sense disambiguation technique was used to determine appropriate sense for each term. To evaluate the performance of their approach, the researchers used Precision, Recall, and F-measure as their evaluation measures. Their approach of Subject-Semantic Clustering (SSA) with Bisecting K-Means achieved precision value of 10%, recall value of 20%, and an F-measure value of 40% more than SSA. Finally, the main advantage of their approach was the reduction of clustering time, as it was based on the declared subjects of the documents.

A simplistic clustering process that involved the use of K-Means and Ant Colony Optimization (ACO) algorithms was shared by Vidhya & Vaijayanthi (2014). The study talked about running the test on two traveling salesman problems and achieving good results by the algorithms. However, the study did not share any tangible information about those results. In general, the study revolved around the broad idea of using document clustering algorithms for DF.

Pallavi, et al. (2014) proposed an approach that applied K-Means algorithm to cluster documents, and based on their experiment showed a performance improvement. They pointed out that the K-Means algorithm they had used had achieved good results when it was properly initialized. Additionally, the results suggested to them that the use of the file names along with the document content information might be useful for cluster ensemble algorithms.

Furthermore, it suggested that clustering algorithms tended to induce clusters formed by either relevant or non-relevant documents, which would help enhance the DF examiner's job. Finally, their evaluation of the proposed approach was on five real-world applications, which showed

that it had potential in speeding up computer examinations. The focus of this study was on the K-Means algorithm; however, no real results were shown to the reader as a point of reference to the achieved results.

Gholap & Maral (2015) proposed a forensic analysis approach to discover useful information in documents. The approach evaluated six different clustering algorithms, including K-Means, K-Medoids, Cluster-based Similarity Partitioning Algorithm (CSPA), and Hierarchical clustering with Single, Complete, and Average Links.

The approach consisted of two phases; the first was used to reduce dimensionality by preprocessing the data, while the second ran the different clustering algorithms. The researchers indicated that their approach had been evaluated using five different real-world investigations. However, no data was shared on those tests.

On the other hand, they explained that based on those results the Average Link and Complete Link algorithms gave the best results. Additionally, K-Means and K-Medoids gave good results when they had a good initialization. Furthermore, they suggested based on the results that combining document content information with filenames might be helpful for cluster ensemble algorithms.

In 2015, Jaybhaye (2015) reviewed a number of document clustering algorithms to show their potential, including the K-Means, K-Medoids, the Expectation Maximization (EM), Hierarchical clustering (Single, Complete, Average Link), Naïve Bayes (NB), and CSPA algorithms. The study concluded that it was hardly possible to get a general algorithm that would work best to cluster all types of datasets, and that the clustering was still an open problem even with the existence of current algorithms.

Rathod & Patel (2015) provided an overview of document clustering, an overview of the forensic analysis process, and surveyed different document clustering techniques used in forensic analysis. The study concluded that data and document clustering were not an easy step to undertake when it comes to computer forensic analysis, because of the various data that would need to be clustered.

Fahdi, et al. (2016) investigated the role of unsupervised pattern recognition of notable artifacts to speed up the DF analysis process. The study attempted to automatically cluster notable artifacts using the Self-Organizing Map (SOM) algorithm. The experiment results showed that the application of SOM for identification of artifacts worked with good performance levels. It was able to correlate notable artifacts using metadata, which led to the identification of 38.6% of notable files, with only 1.3% of noise files.

The experiments ran on four forensic cases; two public and two private. Because of privacy and legal reasons the dataset was limited to these cases, as the researchers had to sign Non-Disclosure Agreements (NDA) in order for them to gain access to the private cases. This stresses the level of difficulty of obtaining real world criminal datasets for testing purposes.

A number of Data Mining algorithms were evaluated by Frederick & Christiana (2017), to find the best ones that could be used to establish the relevance of digital devices in a criminal case without an in-depth forensic examination. The researchers selected four classification algorithms to evaluate and compare their performance.

The selected algorithms were chosen based on performance results obtained by different cited authors. The work evaluated the performance of k-Nearest Neighbor (kNN), Neural Networks,

Bayesian Networks, and Decision Tree, all with a 10-fold Cross-Validation. The learning accuracy performance evaluation indicators were Precision, Recall, and F-measure. Their results indicated that kNN and Neural Network Classifier gave a better performance, given appropriate measures to mitigate the effects of overfitting imbalance in the dataset, than other classifiers.

In 2018, a prototype system was created by Bolle & Casey (2018) to test their proposed approach for finding similarity and links across cases in cyber investigations based on distinctive digital traces. The evaluation for the usefulness of those near similarities was achieved by using data from 207 real world cyber crime cases, which were provided by the State Police in Geneva.

The cases' extracted data was integrated into a MySQL database and searched for links between the cases. The used model allowed for the creation of links between entities using different characteristics, such as email addresses, postal addresses, pseudonyms, or partial identities.

The evaluation focused on email addresses to compute similarity between cases and linking of cases based on the exact similarity between technical characteristics, by using string similarity algorithms on those addresses. Based on the selected threshold of 0.44, the system computed 15,400 comparisons between email addresses. There were 597 address pairs, which led to positive results. However, manual verification by the researchers revealed that only 40 were truly similar.

Malicious software (malware) is frequently used to commit cyber crimes. Therefore research into malware is linked to digital forensic investigations. The advances in malware creation make it harder to be detected, for example through the use of encryption, obfuscation or anti-debugging

techniques. One of the limitations of static analysis of malware is the inability of standard tools, such as anti-virus programs, of detecting code obfuscation.

In 2018, Burnap, et al. (2018) presented an approach that used Self-Organizing Feature Maps to create unsupervised clusters of similar behavior, which were used later as features for classification. The evaluation of the proposed platform was done using a dataset of 1,188 files, 594 of which were malicious files of 32-bit Portable Executable (PE) format and 594 files were benign.

The study relied on behavioral features for each of the files from the dataset after running it in proposed platform including CPU system use, CPU user use, RAMS use, received and sent packets/bytes, and the number of running processes. Those behavioral features were transformed into a feature vector along with the class label, i.e. if they belonged to a malicious or benign file.

The tested machine learning algorithms were Decision Trees, SVM, Probabilistic Bayesian, and Neural Networks. The results were compared to previous research and demonstrated that the presented approach performed better on an unseen dataset ranging between 7.24% and 25.68% in classification accuracy over the other used classification approaches.

Determining the political orientation of a body of text might not be the main concern of DF.

However, there are many benefits that could be found in such a domain if linked to a DF investigation. Abooraig, et al. (2018) collected, manually labeled Arabic articles from different political orientations, and then compared the performance of different feature reduction methods on the dataset.

The best compared methods were the traditional Text Categorization (TC) and the Stylometric Features (SF) approaches. The results of the comparison showed that the TC approach was more superior to the SF approach; TC achieved 90.17% while SF achieved 87.33%. The highest accuracies were reached with the SVM classifier when the Partition Membership (PM) technique was used as the feature selection method.

2.1.3. User Profiling & Other Solutions

Profiling a computer user's gender or age might simplify and speed up the job of finding a suspect in an investigation, and there are different methods to achieve that. For example, using a person's voice recording, or the way a person types, which is known as keystroke dynamics.

A data mining method was introduced by Iqbal, et al. (2008) to identify the authorship of emails. Text mining traditional applications aimed to extract general trends that were found in the text. However, their purpose was to establish frequent pattern features as a way to differentiate the writing styles of different individuals, to determine the author of malicious emails, and to extract supporting evidence to the conclusion reach on authorship.

Thus, they introduced the AuthorMiner method and evaluated it using the Enron⁴ E-mail Dataset, with a randomly selected number of employee's emails from the dataset. The authorship identification accuracy of their proposed method spans an average of 77% to 90%.

⁴ The Enron database contained over 600 thousands emails from 158 employees (from the Enron Corporation) https://www.cs.cmu.edu/~./enron/enron_mail_20150507.tar.gz

Decherchi, et al. (2009) gave an overview of the possibilities offered by clustering-based text mining techniques in the context of a DF analysis. Enron email dataset was used during the experiments to test the proposed methodology. The study proposed a two steps process, the first was based on textual information extraction and the second was textual data analysis through clustering-based text mining tools. The main idea was to provide the analyst with clusters which included documents that were semantically related as a starting point to determine investigation paths.

They conducted a study on the application of clustering text mining techniques during digital investigations for text analysis purposes. The work was said to have used an adaptive model that arranged unstructured documents into content-based homogeneous groups.

In order to simulate an investigational context during tests of the solution, the Enron dataset was used and five randomly selected emails out of a total of 158 authors. The K-Means clustering algorithm was applied only on the body of each email, and the experiments were aimed to test the effectiveness of the approach on two different issues: information retrieval and authorship.

The researchers explained that based on the outcome of these experiments, it led them to believe that the results were influenced by the stylistic metric, which proved to be effective in improving performance accuracy. They further clarified that using 100 clusters, an acceptable accuracy level of 70% can be reached; which was two orders of magnitude less than the total number of emails.

In 2010, Iqbal, et al. (2010) built on their previous work to predict authors in the Enron email dataset using stylometric features, such as sentence length and word length. Three clustering algorithms were evaluated over three experiments for the proposed method; namely,

Expectation Maximization (EM), K-Means, and Bisecting K-Means.

The K-Means gave the best accuracy, from 0.73 to 0.88, when the number of emails per user was limited to 40. However, when more authors were added to the experiments the Bisecting K-Means F-measure score increased from 0.75 to 0.91.

In 2014, Varma, et al. (2014) presented the LIFTR system, which augments the results recovered by the recovery engines it works in concert with. LIFTR was said to prioritize information recovered from Android phones, where a forensic image was extracted by the recovery engine. The suppliers of the forensic image data were three recovery engines; namely DECODE, Bulk Extractor, and Strings (a known UNIX utility that recognizes printable characters of strings in a file).

Recovery engines usually return many unrelated items to the investigated dataset, as it does not consider the nature of the recovered content. The basic idea behind LIFTR was to have all the information recovered through the engines be ranked using different aspects, including the examiner's feedback, the location information of where the files were stored on the system, and the file content itself.

In order to test the validity of the approach, the LIFTR's ranking algorithm was evaluated against 13 refurbished Android phones. The role of the examiner was to label the relevant information items at the page level and provided feedback that was relied on as the foundation of information prioritization. The experiment results, according to the researchers, was that ranking with LIFTR improved the score of the standard information retrieval metric from 0.0 initially to 0.73, which increased after 5 rounds of feedback to an average of 0.88.

Tsimperidis, et al. (2015) investigated the feasibility of identifying user gender using the way they type, and not from the written text content. They used three classifiers to conduct their analysis and achieved 75% accuracy. In order to validate that gender identification was truly language independent, a publically available dataset that contained keystrokes was used.

Although their tests were conducted on a limited number of participants and languages, it still showed the practical implications of this idea. It might be possible to leverage user keystrokes to create user profiles in the context of an information security system or a digital investigation.

The study by Ishihara (2017) compared three different procedures to test the Likelihood Ratio (LR) framework in forensic authorship analysis. One procedure was based on the multivariate kernel density (MVKD) formula, while the other two used n-grams based on characters and word tokens.

The dataset used was a sample of pedophiles and undercover US police officers predatory chat log messages from 115 authors. The dataset was used to see the effects that different number of word tokens would have on the performance of a forensic text comparison (FTC) system. The study demonstrated that the MVKD procedure, with authorship attribution features, performed best in terms of log-likelihood-ratio cost, a metric for LRs quality.

Tsimperidis & Karakos (2018) attempted to predict the gender of a person based on their keystroke dynamics, which could be translated into tens of thousands of features. They assembled a dataset by recording the daily usage of computers by different users, calculated useful features, and trained a few classifiers. The results showed that an unknown user's gender

could be identified with only a few hundred features with an accuracy of over 95%, which according to the researchers was the highest accuracy reported to date in this field.

In 2018, Martinc, et al. (2018) proposed a logistic regression classifier with the aim of discovering gender and variety language from a tweet corpus. Their approach consisted of preprocessing of tweets, feature construction, feature weighting, and the construction of a classification model. The main features of the Logistic regression classifier were different types of character and word n-grams.

The researchers used different n-gram features for the final model, such as word unigrams, punctuation trigrams, and word bigrams. Furthermore, other features included emoji, document sentiment information, and language variety word lists. The best results they achieved in both gender and language variety prediction tasks were on the Portuguese test set with 0.86 and 0.99 accuracy, respectively. It should be noted that the worst accuracy their model achieved was on the Arabic test set.

The paper by Bayne, et al. (2018) highlighted the insufficiency of current DF tools when it comes to their analysis performance on consumer hardware. It proposed a framework for pattern matching of data in a DF context and presented an open-source implementation, called OpenForensics⁵, using an asynchronous Graphic Processing Unit (GPU) solution.

The performance of the developed tool, OpenForensics, was tested and compared to two commercial file carving tools, namely Foremost (v.1.5.7) and Recover My Files (v.6.1.2).

⁵ The open source was made available online on: <https://github.com/ethanbayne/OpenForensics>

OpenForensics achieved the best rates by processing close to 98.7% of the sequential read performance of the measured storage device. Based on the achieved results of the study, OpenForensics was able to perform pattern analysis at a storage device's maximum theoretical sequential read speed, among all the others.

2.1.4. Summary of Digital Forensics Review

Each of the reviewed studies provided a different perspective on what would work best in a digital forensics setting. The noteworthy research directions were as follows:

2.1.4.1. Data Reduction & Triage

1. The use of metadata information was proposed by both Rowe & Garfinkel (2012) and Mohammed, et al. (2016). Rowe & Garfinkel (2012) concluded that file clustering was the most useful technique to find anomalies, but it was challenging for their tool to find deception clues. However, it did find some concealment attempts on some of the samples. Their study showed that using metadata and file clustering could be useful for DF investigations. On the other hand, Mohammed, et al. (2016) proposed a semantic web-based framework for metadata forensic analysis to identify possible evidence from multiple sources. The current research used file clustering as part of the proposed solution. However, the use of metadata to find suspicious files could be an idea to further the study in the future. A drawback of relying on metadata is that it could be directly manipulated by users. For example, a user could manipulate timestamps, edit registry information, or simply rename the files to hide their activities.

2. Roussev & Quates (2012) proposed a content-based forensic triage approach using similarity digests. This method would allow a forensic investigator to screen the content of the data quickly, form a preliminary understanding of the case, and prioritize which devices they ought to commit resources to first. However, it would not directly solve the retrieval overhead problem during the analysis of a selected device.
3. Bharadwaj & Singh (2018) used a random sampling method and sector hashing to find traces of target data. It was observed by the researchers that an increase in the size of the target data with an exponent of two would result in the decrease of the random samples' quantity by two. In other words, if the processing of a lower number of random samples was needed, then the performance of the proposed method might deteriorate. Furthermore, the proposed method could be hindered if the target data were to reside in the last regions of the disk and a sequential method was used, as it would take a longer time to reach it.
4. Quick & Choo (2018) proposed a data reduction framework that added extra stages to the common digital forensics framework. The results indicated that processing of the tested full forensic image took about 8 hours on average, while the logical image that was taken with the data reduction method took approximately 14 minutes on average. An advantage of the proposed method is that common digital forensic tools could be used to apply it.

2.1.4.2. Text Data Mining (TDM)

1. The Self-Organizing Map (SOM) algorithm was used by Fei, et al. (2005) as a way to interpret and visualize similarities in data. The visualization was said to enable forensic examiners to find information of interest more efficiently, but the experimental results lacked details to explain that efficiency.

2. SOM was used also by Beebe & Clark (2007) and Beebe, et al. (2011) to thematically cluster text string search results. They reasoned that these techniques had a higher probability of success in a digital forensics investigation.
3. Another study that focused on SOM was Fahdi, et al. (2016). The interesting aspect of this research was that the only interaction required on the side of the forensic investigator would be when they needed to select the crime category. In addition, this study has the potential to be a foundation for a forensic triaging tool for less complicated investigations.
4. Alternatively, Burnap, et al. (2018) used Self-Organizing Feature Maps to detect Malware using behavioral features that were transformed into a feature vector along with machine learning algorithms.
5. The K-Means clustering algorithm was used by different researchers for document clustering; for instance, Alkoffash (2012), Pallavi, et al. (2014), and Gholap & Maral (2015).
6. On the other hand, Thilagavathi & Anitha (2014) used a hybrid approach with Subject-Semantic Clustering (SSA) with Bisecting K-Means, which achieved a faster clustering time and better accuracy results. Additionally, Vidhya & Vaijyanthi (2014) used K-Means and Ant Colony Optimization (ACO) algorithms for their tests, but did not share the test results.
7. Various researchers reviewed the different clustering algorithms and concluded that clustering was still an open problem and that it was not an easy step to employ when it comes to digital forensics analysis. For example, Jaybhaye (2015) and Rathod & Patel (2015).
8. Other researchers reviewed the different data mining algorithms without an in-depth forensic examination, by establishing a digital device's relevance to the criminal case. Frederick &

Christiana (2017) concluded that kNN and the Neural Network classifiers gave the best performance.

9. Bolle & Casey (2018) looked at the concept of similarity and the use of distinctive digital traces to establish links across cases during cyber investigations. The manual verification by the researchers revealed that from the 597 address pairs that led to positive results, only 40 were truly similar. Forensically speaking, it is essential to confirm that near matches actually correspond to a real relation between found characteristics and to insure that found similarity supports the hypothesis of the links between the different cases.
10. Abooraig, et al. (2018) used Arabic articles from different political orientations to test and compare reduction methods, including the Text Categorization (TC) and Stylometric Features (SF). Arabic news articles as a dataset are a useful resource to test the performance of different techniques. The SVM classifier gave the highest accuracies, while the TC approach was better than the SF approach.

2.1.4.3. User Profiling & Other Solutions

1. Iqbal, et al. (2008) and Iqbal, et al. (2010) used clustering algorithms and stylometric features to identify authorship, while Ishihara (2017) used n-grams for authorship analysis.
2. Decherchi, et al. (2009) used clustering based techniques to provide an investigator with semantically related documents as a starting point to determine an investigation path. They targeted two issues, information retrieval and authorship. While the average accuracy might not have been very high, i.e. 70%, it should be noted that the dataset it was tested against contained short paragraphs of a few words, as it was a real-world Email based corpus.

3. Tsimperidis, et al. (2015) and Tsimperidis & Karakos (2018) used keystroke dynamics to predict the user's gender, while Martinc, et al. (2018) used a logistic regression classifier and different n-gram features for the same purpose.
4. The LIFTR system for information prioritization presented by Varma, et al. (2014) relied on the learnt rules from explored Android phones filesystems and the examiner's feedback. The ranking of the information was based on the examiner's feedback, the information storage location on the system, and the actual content. This was one of the few studies that actively employed the examiner's feedback into the analysis of a device.
5. A framework for data pattern matching was proposed by Bayne, et al. (2018). The study highlighted the failure of digital forensics tools on performing analysis on consumer hardware. Thus, it suggested the use of an asynchronous Graphic Processing Unit (GPU) solution instead, which gave better results during their tests.

2.1.4.4. Research Gap

Although the use of text data mining and the other proposed techniques are reasonable solutions to dealing with the ever growing amount of data that needs to be examined by an investigator. There were few studies with regards to the active utilization of the digital forensics investigator's relevancy feedback to speed up the analysis and reduce the high information retrieval overhead. According to Adelstein (2006), an analysis process during a digital forensics investigation increases proportionally as the volume of data increases.

The use of triage as a solution to the problem would help speed up the forensic task within a shorter period of time and save up on the different resources. The triage approach would be

helpful when the most important pieces of evidence could be extracted and presented to the investigator within the limited time constraints. However, one of the biggest concerns with random sampling, hashing, and triaging is the probability of high false negative rates.

In a digital forensics investigation it might be acceptable to have false positives, even if it would require more effort on the investigator's side. However, an increase of false negatives might change the course of an investigation by allowing potential evidence to slip by undetected, which is the case with current tools that use hashing and string search to look for the occurrence of duplicate files or specific terms.

Data reduction was another solution to help reduce the time and complexity of processing evidence in a digital forensics investigation. Time reduction is one of the main advantages of this approach, as it involves the selective imaging of important files, such as system registry files, Internet browser artifacts, event logs, and other operating system files. On the other hand, the focus on these specific files ignores the user's data. Therefore, data reduction as a solution would be useful in certain types of cases only.

The above methods would work well in a digital forensics investigation in certain circumstances. For example, if an extreme time limit was imposed on the investigator or if they had multiple devices to examine in a short timeframe. Thus, random sampling, hashing, triaging, and data reduction might work well as a discovery tool at the beginning of an investigation to battle the issue of the large volume of data and help in the prioritization of multiple device examinations.

The use of text data mining techniques in general had shown a lot of promise and potential. As highlighted by (Beebe & Clark, 2006), text data mining techniques can help in the extraction of

useful information from data with less processing time and resource constraints. It might also help in the discovery of trends that could go unnoticed by a human.

There were many studies that suggested the use of different text data mining techniques as a solution to the listed digital forensics challenges, as shown previously. However, within a digital forensics investigations setting, the identification of targeted information regarding each case might be an indispensable step for most investigations. And while user profiling or identification might not be directly related to the high information retrieval overhead problem, the ability to identify body of text based on certain characteristics of an author or their gender would be an added advantage to speeding up the analysis of a case.

There is a similarity between the current study and the work by Decherchi, et al. (2009).

Decherchi, et al. (2009) suggested providing the analyst with clusters and assumed that it would present them with investigative clues to continue their work.

There is an opportunity in this stage of the process that had not been exploited enough. In other words, taking the feedback of useful information from the forensic investigator and incorporating it automatically into furthering the analysis process. The current study advocates the use of relevancy feedback from the DF investigator during the analysis phase and regards the use of document clusters as a starting point.

A forensic investigator could supply a system with selected data that constitutes relevant information, which would be useful during a current analysis cycle or as stored intelligence for future cases. The overall goal of this work was to demonstrate the practical use of the digital forensics investigator's feedback after the clustering stage as input for the classification

component, as explained in Chapter Five: Proposed Framework.

2.2. Text Data Mining (TDM)

The two main Text Data Mining (TDM) areas that this research focused on were Document Clustering and Document Classification. TDM can be used to solve different problems and depending on the nature of those problems a supervised or an unsupervised learning algorithm could be used. (Salloum, et al., 2018)

A well known problem solved by TDM using unsupervised learning is document clustering. This method is useful when there are no predefined classes, and the goal is to find certain grouping between similar documents. Document clustering is useful in applications like Information Retrieval (IR).

Another interesting problem for TDM is document classification, which could be solved using supervised learning. In the supervised learning the classes (categories) would be known in advance for each training document. Document classification can be useful in many applications, such as language identification.

The following sections share the literature review on some of the popular clustering and classification algorithms. The aim of this review was to select the most popular algorithms for documents clustering and classification, which would help in the selection of the algorithms used by the proposed framework's components.

2.2.1. Clustering

Clustering is a technique used to partition a set of objects into groups. It can be traced back to the

end of the 1960s, and the early works of Salton (1968) on the efficiency of document clusters to improve search results. (Russell & Norvig, 2010)

Tan, et al. (2005) defined Cluster Analysis as a technique that “groups data objects based only on information found in the data which describes the objects and their relationships”. The goal of the algorithm would result in similar or related objects being grouped together, and different groups would be more distinct from one another based on the efficiency of the used algorithm. The Cluster Hypothesis states that “closely associated documents tend to be relevant to the same requests”. (Rijsbergen, 1979)

The empirical research produced by many previous studies (Hearst & Pedersen 1996; Zamir & Etzioni 1998; Leuski & Allan 2004) showed that the query results using clustering algorithms improved the effectiveness of IR systems over the ones that used relevance ranking models.

In 2007, the work by Ghwanmeh (2007) used Hierarchical K-Means clustering (HKM), which is K-Means-Like with a hierarchical initial set, and found that increasing the number of clusters was not necessary to enhance precision. The clustering ran with a different number of clusters of 2, 3, and 5. The tests ran on 242 Arabic abstract documents that were taken from the Saudi Arabian National Computer Conference. The results indicated that using HKM with 3 clusters outperformed HKM with 2 clusters, HKM with 5 clusters, and traditional IR in terms of precision ranging between (1% - 13%) improvements.

Al-Sarrayrih & Al-Shalabi (2009) applied a Frequent Item set-based Hierarchical Clustering (FIHC) algorithm on an Arabic dataset. The dataset was built in-house and contained 600 Arabic documents. It consisted of 6 categories; including Economics, Politics, Health, Science, Art, and

Agriculture. The tests relied on a different number of clusters, word level n-grams, and character level Tri-grams and Quad-grams. The results were compared to the work done on European languages which achieved 0.62, while the used method reached 0.70.

In 2011, the K-Means clustering algorithm was evaluated by Al-Omari (2011) on Arabic documents and the effects of stemming were estimated by the study. The experiments ran on a dataset of 1,445 Arabic documents that consisted of 9 categories of different topics, such as Economics, Medicine, and Sports. The initial K values varied the results of the experiments from low to very good. The best accuracy score without stemming was 69%, while the best score with stemming was 55%. They attributed these results to the nature of stemming, which might lead sometimes to miss-discriminating of documents.

Nirkhi, et al. (2015) explored the application of unsupervised machine learning methods to authorship verification. They used Hierarchical Clustering and multidimensional scaling techniques. Their aim was to compare similarity between unknown documents against known documents, and using different features would be able to conclude if the unknown documents were written by the same authors.

The evaluation was performed on Enron⁶ corpus using only four authors. Accuracy information of the results was not shared, and the researchers asserted that a range between 70-90% of accuracy in the initial phase would have been acceptable.

Three clustering techniques were tested in the study by Al-Anzi, et al. (2016) and the similarity

⁶ The Enron database contained over 600 thousands emails from 158 employees (from the Enron Corporation) https://www.cs.cmu.edu/~./enron/enron_mail_20150507.tar.gz

results were evaluated. The tested techniques were K-Means, K-Means fast, and K-Medoids. The used dataset was manually collected and comprised of 63 projects from the library of the College of Computer and Information Sciences, Kind Saud University, Riyadh. The best performance was found using K-Means and K-Medoids with cosine similarity.

In 2016, the use of clustering techniques, Latent Semantic Indexing (LSI), Singular Value Decomposing (SVD) methods was proposed by Al-Anzi & AbuZeina (2016). The aim was to group unlabeled similar documents into a number of pre-specified topics. The study relied on an Arabic dataset that was built in-house using 1000 documents of 10 different categories, which contained more than 100,000 unique words. The dataset was collected online from Alanba newspaper website in Kuwait. The tested clustering algorithms included Expectation Maximization (EM), Self-Organizing Map (SOM), and K-Means algorithms. The results indicated that EM clustering gave the best performance over the others; with an average 89% of categorization accuracy. The results of the study indicated that the proposed techniques could be used to label documents without previously shared training data.

Skabar (2017) presented a general graph-based method to cluster a mixed-attribute dataset. The proposed method of the study did not require any explicit measure of distance or similarity and was tested on seven publicly available datasets.

The graph-based algorithm was compared to two relational clustering algorithms; Relational Fuzzy C-Means (RFCM) and Symmetric Nonnegative Matrix Factorization (SNMF). The evaluation of the proposed method indicated that it achieved better or equal to the others' performance on three of the datasets, and achieved second or third best on the remaining datasets.

An approach was presented by Blokh & Alexandrov (2017) to cluster news data using ontology based similarity. The approach relied on semantic similarity metric, which was based on WordNet to find similarity estimation between news messages. The evaluation was done on 415,000 news messages collected from Facebook's official mass media pages, which contained news messages from different news organizations such as CNN, NBC News, NY Times, etc. The results indicated that messages can be grouped into thematic clusters.

The work by Vallejo-Huanga, et al. (2017) presented two semi-supervised clustering algorithms. The first was Clustering Algorithm with Size Constraints and Linear Programming (CSCLP), which relied on two methods to choose the initial points for the clustering, namely the Farthest Neighbor and Buckshot algorithms.

The second was the K-Medoids algorithm with Size Constraints (K-MedoidsSC), variation of the original K-Medoids algorithm. The evaluation of the proposed algorithms showed their validity during experiments that ran on three datasets taken from the University of California Irvine's Machine Learning Repository.

Sardar & Anasari (2018) proposed a parallel k-means algorithm using MapReduce for document clustering. The traditional K-means algorithm was modified into parallel K-means using MapReduce paradigm. The algorithm was designed to allow clustering datasets in short spans of time on top of Hadoop.

The algorithm was tested through experiments that were carried out on a large dataset of newsgroup documents of different categories. It consisted of different sizes ranging between 100 to 1024 megabytes. The end results indicated that the proposed implementation was more

efficient when clustering larger datasets than smaller ones, and in terms of execution time it outperformed sequential K-means.

2.2.2. Classification

Classification is a data mining task that allows a user or another system to come up with predictions based on the knowledge extracted from existing data. Classification of documents of different languages, like Arabic and English, had been applied using different classification algorithms including the K-Nearest Neighbor (kNN), Naïve Bayes (NB), and Support Vector Machine (SVM).

In 2001, a study was conducted by Sawaf, et al. (2001) that showed that document clustering and classification was possible for Arabic text even with no morphological analysis. The study used statistical methods on the Arabic NEWSWIRE corpus. The news articles covered different topics, such as politics, economy, and sports. The highest maximum entropy text classification F-measure that was reached was 62.7%.

In 2004, Naïve Bayes (NB) was used for the automatic classification of 1500 Arabic text documents by El-Kourdi, et al. (2004). The study used cross validation experiments on a dataset that consisted of five categories that had 300 web documents per category, and depended on different root extraction algorithms for 2,000 Arabic terms/roots. The results showed that the average accuracy of all categories was 68.78%. This study offered an indication that classifying Arabic documents with the NB algorithm is not affected by the Arabic root extraction algorithm. As indicated by the comparison of their results with previous similar research results that did not use such an extraction algorithm.

The k-Nearest Neighbor (kNN) with a Document Frequency Threshold method was implemented by Al-Shalabi, et al. (2006) for Arabic text categorization. They used a dataset that consisted of 621 Arabic text documents and scored for precision and recall a micro-average of 0.95. The results of this study indicated the adaptability of the kNN algorithm to Arabic text categorization. Mesleh (2007) applied a Support Vector Machine (SVM) algorithm with Chi Square feature to classify Arabic articles. The experimental study used an Arabic corpus of 1,445 online articles from different sources and classified them into 9 categories. The results gave an average of 88.11% of F-measure.

The work by Gharib, et al. (2009) compared the performance of the Support Vector Machine (SVM) algorithm in classifying an Arabic dataset and compared its performance with Bayes, K-Nearest Neighbor (kNN), and Rocchio classifiers. The experiments were conducted on a dataset that contained 1,132 Arabic documents, which were collected from 3 different Egyptian newspapers and consisted of 6 news categories, such as Sports, Economics, and Technology.

The SVM classifier gave the best accuracy results over all the others in high dimensional feature spaces. For example, when using more than a 4,000 feature set the classification rate exceeded 90%. However, SVM was the most expensive as it required more time to finish than the others, while the Bayes classifier was the most efficient in terms of time.

Alsalem (2011) used Naïve Bayes (NB) and Support Vector Machine (SVM) algorithms on different Arabic datasets. The dataset contained 5,121 Arabic documents of different lengths and consisted of 7 news categories, such as Sport, Economics, Technology, and Politics. The evaluation measures used were precision, recall, and F-measure. The overall average F-measure

for SVM was 0.778, while for NB was 0.74. The results indicated that SVM gave a better performance than NB on all evaluation measures.

Zrigui, et al. (2012) proposed a new hybrid algorithm based on the Support Vector Machine (SVM) and the Latent Dirichlet Allocation (LDA) classifiers. The proposed model adopted (topics) vector sampled by LDA, instead of relying on an n-grams, i.e. vector of words.

The experiments were conducted on a dataset that was built in-house, which consisted of 1,500 Arabic documents from different online websites and news agencies. It consisted of 9 news categories, such as Sport, Economics, Medicine, and Arts. The evaluation measures used were precision, recall, and F-measure. The results indicated that the proposed algorithm scored better with a Macro average F-measure of 88.1% and Micro average F-measure of 91.4% than the Naïve Bayes (NB) and k-Nearest Neighbor (kNN) algorithms.

In 2013, in order to improve the accuracy of stemming for Arabic text Hadni, et al. (2013) proposed a hybrid method for a Text Categorization (TC) system along with the Naïve Bayes (NB) and the Support Vector Machine (SVM) classifiers. The experiments were conducted on a large Arabic corpus that consisted of 12 categories, such as Economics, Politics, Health, Recipes, Religion, and Sports.

The performance evaluation of the proposed hybrid method was compared with other Arabic stemmers, including a root-based approach (Khoja Stemmer), a stem-based approach (Light Stemmer), and a statistical approach (n-gram). The hybrid method with the NB classifier gave the best results overall the other stemmers. The average F-measure score for the hybrid-NB approach was 70.5%, 67% with the Khoja Stemmer, 51% with the n-gram, and 63.2% for the

Light Stemmer. The current study did not take into account the issue of stemming of Arabic words, but this could be an interesting issue for future development.

In 2016, the work by Mohammad, et al. (2016) discussed the problem of Arabic text classification and used three classification algorithms on a dataset that was built in-house. The tested algorithms were Support Vector Machine (SVM), Naïve Bayesian (NB) and Multilayer Perceptron Neural Network (MLP-NN).

The dataset was a collection of 1,400 Arabic documents from 3 different news agencies and consisted of 9 news categories, such as Sport, Economics, and Politics. The training set contained 880 documents, while the test set contained 520 documents. The average precision scored by SVM was 0.778, by NB was 0.754, and by MLP-NN was 0.717. The results indicated that SVM performed better than all the others.

In 2016, Venturini, et al. (2016) proposed a distributed Bagged Associative Classifier (BAC) model, which was based on ensemble techniques. The classifiers would train their models in parallel by distributing the workload among the machines of a cluster. The classifier was implemented on Apache Spark, and the evaluation of the classifier was performed on three datasets. The results indicated that sampling by itself will not always be sufficient to reach a good accuracy, it is not always feasible to train an associative classifier over the whole dataset, and that bagging offered a practical method to distribute the workload among workers and to reach the quality of a single classifier.

Silva, et al. (2017) proposed a multinomial text classifier named MDLText, which was based on the Minimum Description Length principle. Their aim was to ensure that the classifiers would

achieve a good trade-off between the model complexity and at the same time attempt to avoid overfitting. The evaluation of the proposed classifier performance was done using 45 public text corpora using two scenarios, namely batch learning and online learning.

They also compared the achieved results to other traditional machine learning algorithms, such as Naive Bayes, SVM, and k-Nearest Neighbor (kNN). The results indicated that the proposed classifier outperformed the others in terms of prediction. On the other hand, it did not indicate a significant difference with SVM's achieved results. However, in terms of efficiency the MDL classifier was 56 times faster on average. Their study relied on a principle that states that in a decision between two or more models to fit some data, the less complex models would be preferable.

In 2017, Karystianis, et al. (2017) implemented a generic rule-based method to enable automatic classification of documents; by recognizing any mentions of targeted elements in epidemiological study abstracts. The rules were based on common text syntactical patterns, which were tested on 35 manually selected epidemiological study abstracts. The results showed F-measure of 70-98%, precision of 81-100%, and recall of 54-97%.

The W2VLDA system was presented by Garcia-Pablo, et al. (2018). It performed sentiment classification with almost no supervision or a need of domain or language specific resources. The system combined different unsupervised approaches, such as word embeddings and Latent Dirichlet Allocation (LDA).

The almost no supervision part came from the only requirement of user supervision being a single seed word entered by the user for each desired aspect or polarity. The system performance

was evaluated using customer reviews from several domains using the multilingual ABSA dataset. It was also compared to other LDA-based approaches. The proposed system outperformed all others with a 95% of confidence for all languages, except for Dutch where it only achieved an 80%.

In 2018, ensemble methods were used by Abdelaal, et al. (2018) to improve classification accuracy of Arabic tweets. The dataset was collected from twitter using the Application Programming Interface (API) and resulted in 500 tweets of five different categories, and each tweet was manually labeled according to their contents. The experiment results indicated that using ensemble methods performed better than individual methods. The results showed an increase in the Sequential Minimal Optimization (SMO) classifier of 2.2%, an increase in the Naïve Bayes classifier to 1.6%, and finally an increase of 3.2% in the Decision Tree (J48) classifier.

The work by Alakrot, et al. (2018) presented an approach for the detection of offensive language in online communication in Arabic using a Support Vector Machine (SVM) classifier. The researchers experimented with different preprocessing techniques, word-level features, and n-gram features. The evaluation dataset was comprised of YouTube comments in Arabic. The achieved accuracy was 90.05%, which was better than previous studies that focused on Arabic text.

2.2.3. Summary of TDM Review

The most popular algorithms for document clustering were K-Means (Al-Anzi, et al. 2016; Sardar & Anasari 2018) and Hierarchical clustering (Al-Sarrayrih & Al-Shalabi 2009; Nirkhi, et al. 2015).

Whereas, the most popular document classification algorithms were k-Nearest Neighbor (Al-Shalabi, et al. 2006; Silva, et al. 2017), Naïve Bayes (El-Kourdi, et al. 2004; Abdelaal, et al. 2018), and Support Vector Machine (Mesleh 2007; Alakrot, et al. 2018) and these three algorithms were often tested together.

There were many studies that proposed the use of modified versions of clustering or classification algorithms (Ghwanmeh 2007; Skabar 2017) or used an added feature (Blokh & Alexandrov 2017; Vallejo-Huanga, et al. 2017) or used ensemble techniques (Venturini, et al. 2016; Abdelaal, et al. 2018). However, such advanced techniques could be considered in the future for the development of the proposed framework.

2.3. Relevance Feedback

Over the past 50 years, many techniques were developed that used knowledge about users to improve the performance of document retrieval systems. The aim of the following review was to select the best method of integrating a user's relevancy feedback into the proposed solution.

In the context of an Information Retrieval (IR) system, data retrieval deals with determining which documents from a set contain the keywords taken from the user's query. According to Baeza-Yates & Ribeiro-Neto (2011), in most instances this type of retrieval is insufficient to satisfy the user's request, as the user is more likely to be concerned about retrieving information that pertain to their topic of interest, than retrieving data that only matches the given query.

Data retrieval is useful in many instances, such as during a hunt for data in a relational database, which is known to follow a well structured format. On the other hand, IR systems usually deal

with natural language text, which is often unstructured and could sometimes be semantically ambiguous. (Baeza-Yates & Ribeiro-Neto, 2011)

Information Retrieval systems differ from Information Extraction (IE) and Knowledge-Based Systems (KBS) in that they're not typically used to extract information from objects, or manipulate, or infer information for a user. They can be used instead to lead the user to objects that could be deemed useful.

In the beginning of a digital forensics investigation, not all facts might be known to the examiner of a case. And the more they progress in their analysis, the more facts become known and new relationships between items develop. Similarly, a user of an IR system might not have all pertinent information from the beginning, they might not be able to articulate the idea in a suitable query of what they're looking for, or they might not even have an idea of which information is available to be retrieved.

Additionally, while users might have a difficult time expressing accurately the required information or files, they can sometimes recognize information they deem useful once they see them. That said, a user might be able to indicate which documents or pieces of information has useful information if they were presented with an initial set of documents with potential relevance to their request. (Ruthven & Lalmas, 2003)

For an IR system to be able to achieve its goal of satisfying the user's request for information, it needs to be able to interpret the contents of the documents from a collection and be able to rank them based on their relevance to the user's query. The notion of relevance is at the core of any IR system.

Relevance Feedback is a relatively old concept, as it was introduced in the mid-1960s. The main idea behind it consisted of taking the results that were returned from processing a given user query and providing those results to the user for relevance evaluation. Then, it uses that relevancy information to perform a new, improved, query to achieve better results. (Baeza-Yates & Ribeiro-Neto, 2011)

Past research on Relevance Feedback (RF) in its original form (Salton & Buckley 1990; Harman 1992; Buckley, et al. 1994) had shown that it could be an effective technique to improve retrieval results. As explained by Baeza-Yates & Ribeiro-Neto (2011, p. 4), the retrieval of all relevant documents to a user query is the primary goal of an IR system, while disregarding as many non-relevant documents as possible. A typical query search over a document collection starts with the user initializing their need by specifying a certain query, which is then processed to obtain retrieved documents. This operation is usually made faster by preprocessing the collection using an indexing algorithm.

After retrieving the documents from the collection, a ranking algorithm can be used to rank the selection according to the likelihood of relevance to the user's initial query. Afterwards, the user can examine the selection of ranked documents to look for useful information, and their activities could be used to provide more information for the IR system to further develop future query results. For example, the IR system could rely on explicit feedback from the user to rate the retrieved documents for relevance or it can rely on implicit feedback by gathering information on the documents the user clicked on. (Baeza-Yates & Ribeiro-Neto, 2011)

Query feedback methods are usually used for Query Modification, which is commonly referred to

in the literature as Relevance Feedback or as Query Expansion. Relevance Feedback occurs when the user provides further information on the relevance degree of documents to the user's query. Query Expansion refers to the use of information related to the query in order to expand it to achieve better results.

The Rocchio Algorithm relies on user relevance feedback of retrieved documents. Improvements in precision rates were shown in early experiments using the Smart system by Salton (1971), and using the probabilistic model by Robertson & Jones (1976) when relevance feedback was used on small test collections. A shortcoming of the Rocchio's Algorithm is that evaluation of relevancy by the user is required in the course of multiple iterations to distinguish items as relevant or non-relevant.

A typical feedback cycle, as described by Baeza-Yates & Ribeiro-Neto (2011, p. 178), is composed of the following steps:

1. Determining the feedback information which is related, or expected to be related, to the original query. This can be obtained either:
 - a. Explicitly: from the user, either directly by the user or by a group of human assessors.
 - b. Implicitly: from the query results or from external sources, i.e. Thesaurus. There is no participation of the user in the feedback process.
2. To take this information effectively into account, determining how to transform the query, this can be achieved through a variety of methods.

Salton & Buckley (1990) pointed out many advantages of the RF procedure. For example, the user

is shielded from the process details for the query formulation, the search operation is broken down into a sequence of search steps, and based on the search environment it helps emphasize and deemphasize some terms by controlling the query alteration process.

There are different approaches for RF, including the Boolean Model, Vector Space Model, and the Probabilistic Model. The Boolean Model is the simplest as it is based on the matching between query and documents.

There are two possible methods to implement RF on Boolean systems as suggested by Harman (1992). The first method is to present a list of possible new query terms to the user, which can be chosen based on term distribution in the relevant documents. The second method is to automatically modify Boolean queries by the system. An example of such a system was proposed by Khoo & Poo (1994), which was designed to automatically modify the Boolean connectives of the queries and the used terms based on the documents that were marked as relevant by the user. (Ruthven & Lalmas, 2003)

In the Vector Space Model (VSM) each document is represented by a vector and the documents that were identified as relevant to a given query have similarities between them. The basic idea, as described by Baeza-Yates & Ribeiro-Neto (2011, p. 181), is that the query would be reformulated to get closer within the vector space to the neighborhood of the relevant documents and as far away from the non-relevant documents' neighborhood. The main advantages of this model are simplicity and that good results are achieved. As the modified term weights are directly computed from the set of retrieved documents, because a portion of the intended query semantics is reflected in the modified query vector.

In the Probabilistic Model, documents similar to a query are ranked according to the probabilistic ranking principle. Robertson (1997) explained that in the probabilistic ranking principle a system's response to a request is a ranking of the documents in a collection, which is based on the order of decreasing relevance probability to the user's request. Thus, the overall effectiveness of the system would be based on whatever data have been made available to the system.

The main advantage of this model is that the derivation of new weights for the query terms is directly related to the feedback process. However, this model has several disadvantages. For example, during the feedback loop document term weights are not taken into account, it usually disregards the weighting of terms from the previous query formulations, and the query expansion is not used. (Baeza-Yates & Ribeiro-Neto, 2011, p. 184)

This study was not geared towards query modification or expansion. Nonetheless, it is important to understand the underlying concepts behind search requests and the variables that affect optimization of RF.

Most users, especially on the Internet, are unwilling to provide feedback on retrieved search results and often dismiss the whole process once they get what they're looking for. Therefore, it is expensive to obtain information on documents that are relevant to the user's query as it requires the user's direct action. In addition, because of the high cost of figuring out the relevancy score of retrieved information, alternative feedback methods were developed.

For example, looking at which documents were selected (i.e. clicked on) by the user, instead of asking the user if the retrieved information or documents were relevant to their query.

Alternatively, a look at term-frequency of the top documents that were retrieved can provide

good relevancy information. Subsequently, better results might be produced in the future if it was assumed that such information (i.e. documents selected by the user or frequency of top document's terms) was related to the original query.

When a user searches for information the request formulation is a complex one; as it depends on some particulars of the requestor. For instance, the variables that might affect the results include the requestor's knowledge of the contents' store, the topic matter being searched, and personal preferences of vocabulary and style. Consequently, a statistical decision that is based on the user's personal experience must be made in order for them to retrieve useful results based on their query. (Rocchio, 1971)

Furthermore, a user who needs to know if a specific document is within a certain dataset can easily submit a request that is identical to the document in question. On the other hand, a user who needs information on an unfamiliar topic would face difficulty formulating such request. Therefore, operationally speaking, the likelihood of satisfying the user's needs in the context of Information Retrieval systems vary based on a wide spectrum of variables. The optimization of search requests is believed to be comprised of a better control in evaluating request-document matching and indexing. (Rocchio, 1971)

There are two basic steps of a feedback cycle. The first step is to establish which feedback information is either related or expected to be related to the original query. In order to achieve this step, two distinctive methods can be used; either explicitly from the user or implicitly from the query results.

The second step is to establish how the original query could be transformed to effectively take into account this relevancy information. In this study, only the first step's methods are focused on, as the second step is beyond the scope of this research.

Originally, in order to collect explicit user feedback a user would inspect the top retrieved documents generated from their query and would indicate which of those are relevant to their needs. The collected feedback information from different users is only taken into account if it was supported by a majority of the users.

However, in many cases users might be unreliable to judge relevance or might be unwilling to participate. Therefore, another option was to have a group of specialists assess the relevance of retrieved results. It turned out that relying on the judgment of users or specialists was both time consuming and expensive.

Alternative to the above scenarios, a user's behavior on the web can be used to collect relevancy information of query results without disrupting the user's experience. When a user is given a list of ranked document as a result of their query, by collecting information on which of those results were clicked on by the user, one can estimate the probability of their relevance.

According to Baeza-Yates & Ribeiro-Neto (2011, p. 178), the difference between the relevancy selection method by the user and the clicked on results by the user, is that the second method does not present any deviation to the user's search task and makes their participation of this type of feedback natural. A click on a result does not necessarily constitute that a document is relevant to the user's query, but it does indicate that it might be of interest to the user in the context of their search.

The direct participation of the user during the feedback process is not required in the implicit feedback cycle. In this type of cycle, the feedback information is drawn implicitly by the system using different approaches; either through local or global analysis methods. The local analysis relies on the top ranked documents in the retrieved results to derive the feedback information of their relevancy. Whereas, the global analysis approach uses external sources, i.e. thesaurus or using term relations taken from the document collection.

The implicit method's attractiveness is garnered from the low cost of collecting it and that it does not need the involvement of the user. On the other hand, the drawback of this method is that the feedback information might not necessarily be related to the current user's query, as there is no direct input from the user to confirm its relevancy. (Baeza-Yates & Ribeiro-Neto, 2011, p. 180)

In the context of a DF investigation, a search of digital evidence is usually done by forensically analyzing digital devices that typically have a finite number of documents. The main advantage in this case is that it might be easier to receive explicit feedback from analysts who are invested in the betterment of future searches.

Additionally, relevant text data of value can be mined from different cases of the same category and used to optimize future search jobs. For instance, the term frequency in cases such as terrorism, narcotics or money laundering could be collected and classified in a shared database that could be used for different DF investigations.

The explicit feedback method was selected to be used in the proposed framework. The user of the proposed framework would be required to indicate which of the retrieved results are relevant.

Those relevant results would be used during the analysis and would be added to a database, which could be used for future investigations.

Chapter Three: Text Data Mining Background

This chapter explores the concepts behind the Text Data Mining (TDM) components of the proposed framework, including Clustering and Classification. TDM describes a range of technologies for the analysis and processing of unstructured or semi-structured textual data. The text could be a collection of unstructured documents or text mixed with numbers. (Makhabel, et al., 2017)

The goal of all of these technologies could be summed into turning text into numbers, so that algorithms could be used on large document databases. These technologies could be divided into seven distinct areas, based on the unique characteristics of each. A project within TDM could be said to use techniques from multiple areas. (Elder, et al., 2012)

According to Elder, et al. (2012, p. 32) the seven practice areas include:

1. **Search & Information Retrieval (IR)**: includes the indexing, search, and retrieval of documents from large text databases using queries.
2. **Document Clustering**: the use of data mining clustering algorithms to group and categorize text or documents.
3. **Document Classification**: the use of data mining classification algorithms to group or categorize text or documents based on models that were trained on labeled examples.
4. **Web Mining**: text and data mining on the Internet by drawing on methods from document classification and natural language processing, with a special focus on the interconnectedness of the web.
5. **Information Extraction (IE)**: the process of constructing structured data from unstructured or semi-structured text by identifying and extracting relevant facts and relations from it.

6. **Natural Language Processing (NLP)**: low level language processing, i.e. part of speech tagging.
7. **Concept Extraction**: the creation of words' grouping based on the semantic similarity of the groups.

The following sections review background information on the TDM components that comprised the two integral parts of the proposed framework, namely Clustering and Classification.

3.1. Clustering

There are generally two main types of text classification algorithms, unsupervised and supervised. Unsupervised algorithms (Figure 3.1) are mostly used to handle a large collection of text where there is no training data available and requires no human intervention. For instance, when no information is given as input about which documents in a set belong to a previously specified class.

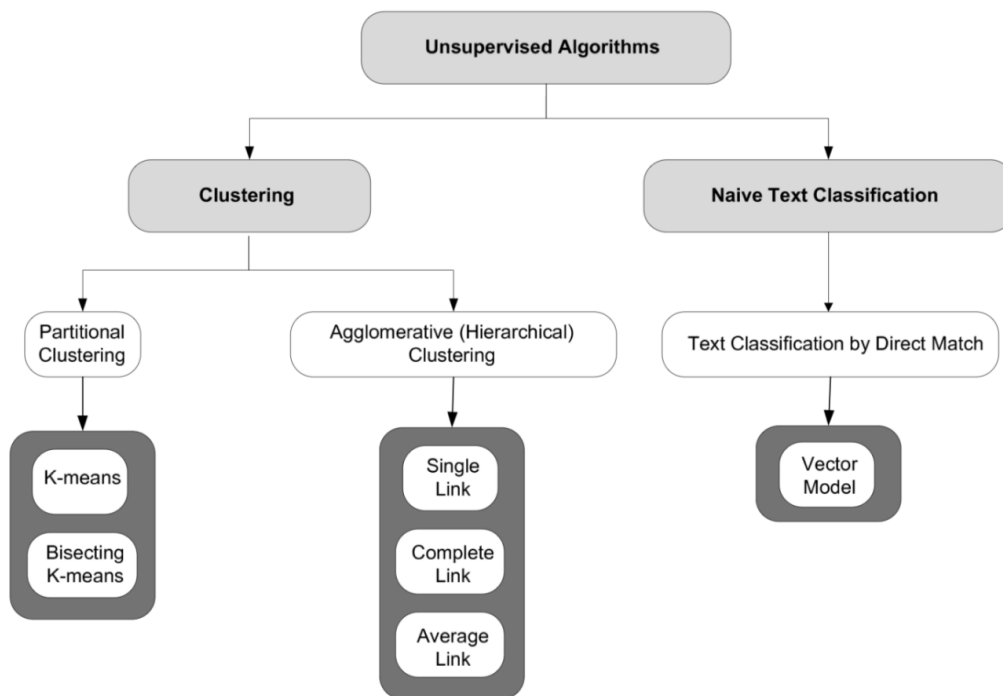


Figure 3.1: Unsupervised Algorithms (Baeza-Yates & Riberio-Neto, 2011, p. 284)

Conversely, supervised algorithms (Figure 3.2) lead to better results, however they require the availability of training data. This type of algorithms uses input data given by humans or by means of human assistance. (Baeza-Yates & Ribeiro-Neto, 2011, p. 284)

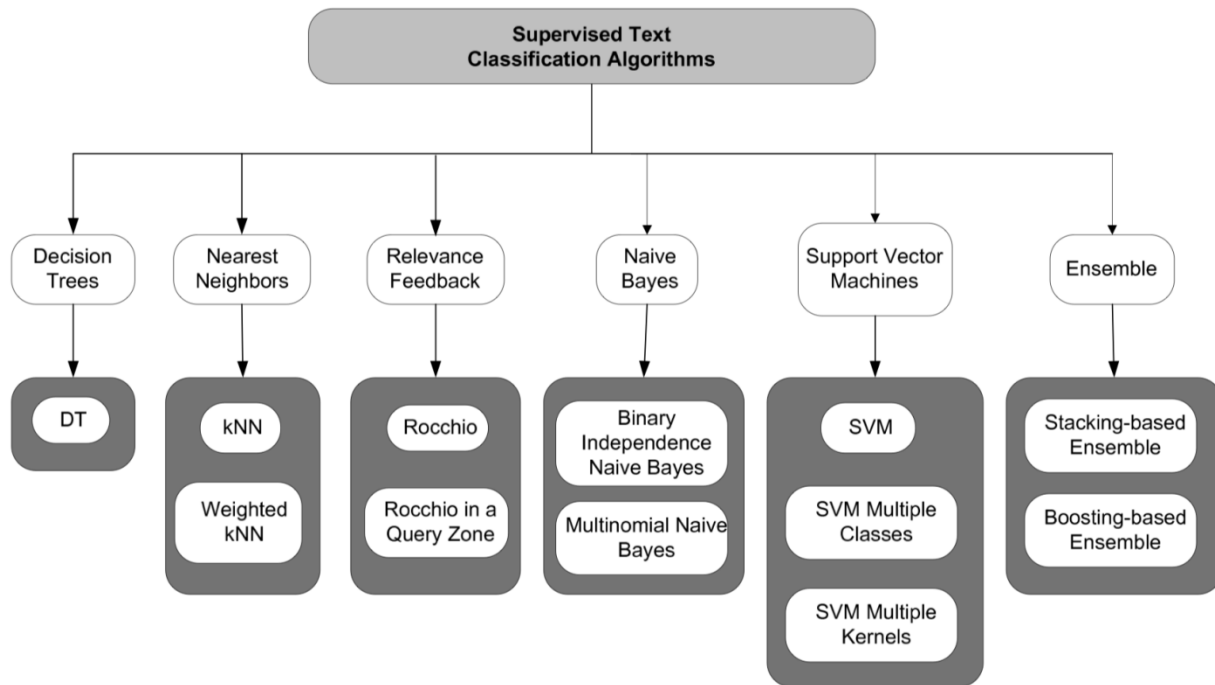


Figure 3.2: Supervised Algorithms (Baeza-Yates & Riberio-Neto, 2011, p. 285)

Clustering techniques are mostly applied when instances need to be divided into natural groups, but there is no class to be predicted. There has to be some aspects of resemblance between the items of a collection for them to be grouped into certain clusters. (Witten, et al., 2016, p. 141)

The objects within each cluster are more closely related to one another than objects assigned to different clusters. Additionally, an object can be described by its relation to other objects, or by a set of measurements. Furthermore, sometimes the goal is to arrange the clusters into a natural hierarchy. That involves grouping them in such a way at each level of the hierarchy, whereby the

clusters within the same group are more similar to each other than those in different groups.

(Hastie, et al., 2016, pp. 501-502)

Regardless of the clustering algorithm used, there is a central concept to the degree of similarity (or dissimilarity) between the clustered individual objects. Objects are grouped by a clustering method based on the definition of similarity supplied to it. (Hastie, et al., 2016)

The unsupervised algorithms are useful in cases where the data classification is unknown in advance, or when it is too expensive for the data classification to be done by a human.

Consequently, large datasets can be reduced through these algorithms into fewer informative representatives.

3.1.1. K-Means Clustering

Back in 1957, a simple iterative algorithm was suggested by Stuart Lloyd (1982), but it was not published outside Bell Labs until 1982. James MacQueen was the first to use the term “K-Means” in 1967, although the idea goes all the way back to 1957 with Hugo Steinhaus. (MacQueen, 2008)

The K-means is a classic algorithm with a Top-Down procedure, and it is known to be one the most popular iterative descent clustering methods. It is used when the dissimilarity measure is chosen as squared Euclidean distance, and all variables are of the quantitative type. (Hastie, et al., 2016)

In the first step, the user specifies how many clusters will be created in advance; this is known as the K parameter. Then, K points are chosen randomly as the cluster centers. Next, the ordinary Euclidean distance metric is used to assign all instances to their closest cluster center.

The “means” part of the algorithm’s name comes from the centroid (mean) of the instances in each cluster. These centroids are then considered the new center values for their clusters. Finally, this whole process is reiterated with the new cluster centers. Those iterations continue until the centers are stabilized and will remain the same. (Witten, et al., 2016, p. 143)

This clustering algorithm is known to be simple and effective. However, small changes in the initial random choice will result in completely different arrangements. This fact is true for all clustering techniques.

People often run an algorithm several times in order to increase the chance of finding a global minimum. Different initial values would be chosen and the best final result would be the one with the smallest total squared distance. In general, finding globally optimal clusters is almost always infeasible. (Witten, et al., 2016, p. 144)

So, how is the best value of K chosen? The whole point of clustering is to find the likely number of clusters; as often nothing is known about it. Therefore, the user of the developed tool should have the option to select the value of the K parameter; in order to reach better results.

The popularity of the K-means algorithm stems from its simplicity and that it is easy to understand and implement. On the other hand, it has a number of weaknesses including the requirement for the user to specify the number of clusters K , its sensitivity to outliers and to initial seeds, it can only be used if the mean is defined, and that the centroid is usually replaced by the most frequent exemplar for categorical data.

3.1.2. Hierarchical Clustering

The choice for the initial configuration and number of clusters to be searched in K-means algorithms determines the results that will be achieved. Conversely, such specifications are not required in hierarchical clustering algorithms. They do require that based on the pair-wise dissimilarities among the observations in two (disjoint) groups, a measure of dissimilarity between them is specified by the user. (Hastie, et al., 2016, p. 520)

Hierarchical representations are created in this type of clustering, whereby the clusters at each hierarchical level are created by merging clusters at the next lower level. Moreover, each cluster contains a single observation at the lowest level, while there is only one cluster that contains all of the data at the highest level. There are two basic strategies for hierarchical clustering: bottom-up (Agglomerative) and top-down (Divisive).

The Agglomerative (bottom-up) method starts at the bottom and recursively merge a selected pair of clusters into a single cluster at each level. As the name suggests, in the bottom-up approach each instance starts in its own cluster and as one move up the hierarchy pairs of clusters are merged. (RapidMiner, 2018)

This means that the two groups of the smallest intergroup dissimilarity are chosen for merging the clusters, so that the grouping at the next higher level will produce one less cluster. The way this works is as follows (RapidMiner, 2018):

1. Put each item in its own cluster.
2. Choose the two clusters with the smallest distance, among all current clusters.
3. Merge the two original clusters with a new cluster.
4. Repeat the last 2 steps, until only one cluster remains in the pool.

A rooted binary tree can be used to represent this recursive binary splitting (Agglomerative). The entire dataset is represented by the root node, while the groups are represented by the nodes of the trees, and individual observations (clusters) can each be represented by terminal nodes.

This type of clustering can be represented graphically using a dendrogram, which is one of the main reasons for the high popularity of this clustering method. A dendrogram is a graphical format that provides an interpretable and complete description of the hierarchical clustering. (Hastie, et al., 2016, p. 521)

Aside from the many choices of distance measures, the user needs to select which of the linkage criterion to use. There are three different linkage strategies within the Agglomerative clustering algorithm, namely single-link, complete-link, and average-link.

On the other hand, the Divisive (top-down) method does the opposite as it starts at the top and recursively splits one of the existing clusters into two new clusters at each level as it moves down the hierarchy. In order to produce two new groups the split is chosen based on the largest between-group dissimilarity.

Hierarchical clustering creates levels, where each level represents a specific grouping of the data into disjoint clusters of observations. The user decides which levels (if any) represent a natural clustering. This is based on the similarity between observations within each group, compared to observations assigned to different groups at the same level. (Hastie, et al., 2016, p. 521)

This algorithm is not without its own set of limitations. According to Tanaseichuk, et al. (2015), it is challenging to apply hierarchical clustering to large datasets. The most popular average-linkage hierarchical clustering requires $O(n^2)$ of time, compared to the linear complexity of the k-

means algorithm. The typical Agglomerative Hierarchical Clustering cannot handle large datasets well, because of the time complexity. (Kaushik, 2016)

3.2. Classification

A Classification algorithm is a process of selecting the best option to fulfill the requirements based on a given scenario. And as the name “Classification” suggests, the task basically involves classifying new cases or examples. Additionally, the training dataset used always contains all the possible classes that could be used. (Rahman, 2012)

There are many popular classification algorithms when it comes to TDM, such as k-Nearest Neighbor (kNN), Naïve Bayes, and Support Vector Machine (SVM). The kNN algorithm compares an unknown dataset (Example) with the K training dataset (Examples), based on the nearest neighbors (closeness is defined in terms of a distance) of the unknown Example. The first step for this algorithm is finding the K closest training (Examples), then in the second step using a majority vote of the found neighbors the kNN classifies the unknown Example. (RapidMiner, 2018)

There are certain drawbacks to this algorithm; for example, its success depends on selecting the best value of K, and when the training dataset size grows it becomes slower and thus requires more time for object classifications. (Cherkassky, 1995)

The Naïve Bayes classifier is a simple classifier that works on numerical and textual data. It is based on a probability algorithm with strong independence assumptions. The fundamental assumption being that given the value of the class, the value of any attribute is independent of the value of any other attributes. The attractive thing about this method is that it estimates the

parameters necessary for classifications by relying on a small amount of training data. On the other hand, this classifier has difficulty dealing with non-relevant features or noise in training data. (Syiam, et al., 2006)

The Support Vector Machine (SVM) is a statistical algorithm and one of the most effective text classification methods. Its main goal is to find the optimal separating hyper-plane, which has the maximal margin to both sides with the lowest true error. (Mesleh & Kanaan, 2008)

The increased number of features provides a clearer optimal hyper-plane. However, this needs a large set for training and a large number of features to work properly. (Hmeidi, et al., 2008)

There is a need to find certain criteria to assess a classifier's performance during experiments.

One of the most common problems for data scientists is the availability of good training data.

Therefore, it became a known practice to use the available data for both training and testing.

Better generalization is achieved with more training data, as having more test data gives far superior estimates for the error probability of classifications. It is said that conclusions reached when evaluating the performance of classifiers on training data alone is positively biased.

(Hlaváč, 2018)

There are many solutions to handle this problem. For instance, Cross Validation (Kohavi, 1995), Bootstrap (Breiman, 1996), and Hold Out (Hlaváč, 2018), as shown in (Table 3.1). During the experimental validation procedures for this study, the K-fold Cross Validation with ($K = 10$) was selected to estimate performance accuracy.

Method: K-fold Cross Validation	
Description	Accuracy Estimation
<ol style="list-style-type: none"> 1. Training dataset is randomly divided into K disjoint sets. 2. The K sets will be of equal size. 3. Each set will have roughly the same class distribution. 4. The classifier will be trained K times. 5. Each time the classifier will be trained with a different set held out as a test set. 	<p>Error estimation is the mean of K errors.</p>
Method: Bootstrap	
Description	Accuracy Estimation
<ol style="list-style-type: none"> 1. This method generates multiple versions of a predictor and uses these to get an aggregated predictor. 2. By making bootstrap replicas of the learning set, the multiple versions are formed. 	<ol style="list-style-type: none"> 1. When predicting a numerical outcome, the aggregation averages over the generated versions. 2. When predicting a class, then plurality vote is used. 3. The instability of the prediction method is a vital element, meaning that if the disruption of the learning set can cause significant changes in the constructed predictor, then the bootstrap method is said to improve accuracy.
Method: Hold Out	
Description	Accuracy Estimation
<ol style="list-style-type: none"> 1. Randomly partition data into 2 independent sets. 2. The Training set will be using 2/3 of data for learning the classifier. 3. The Test set will be using 1/3 of data (hold out) for the accuracy estimation of the classifier. 	<ol style="list-style-type: none"> 1. Hold out is repeated K times. 2. The average of the accuracies obtained gives the estimated accuracy.

Table 3.1: Accuracy Estimation
(Adopted from (Kohavi 1995; Breiman 1996; Hlaváč 2018))

Chapter Four: Research Approach

This chapter explains the research design, reliability, and ethical considerations.

4.1. Research Design

The challenges of high information retrieval overhead and the increase of Digital Forensics (DF) examination time could be attributed to the ever growing volume of data, and the high recall and low precision rates associated with previously existing DF search tools. Few studies in the literature were undertaken with regards to using relevancy feedback from the DF investigator during the analysis phase of an investigation.

There is a wide array of research paradigms and methods that could be used to conduct this study. The nature of a research problem usually influences the choice of methodology that would be most suitable to address it.

According to Saunders, et al. (2012, p. 161), research methods can be classified into Quantitative, Qualitative, or Mixed Methods. Quantitative research deals with data collection or analysis procedures that use numerical data and examines relationships between variables. On the other hand, qualitative research deals with data collection or analysis procedures that use non-numerical data, for instance words or images. Furthermore, based on the nature of the research problem these two approaches might be combined in what is known as the mixed method.

Based on the practical implications of the research problem, different research strategies can be used. For example, the use of the Case Study strategy might be relevant to build a theory based on particular instances of a phenomenon using one or more cases. The idea is to use a case or cases

to inductively develop a theory, which would be developed based on recognized patterns of relationships within a case or among those cases. (Eisenhardt & Graebner, 2007)

Although the use of the Case Study strategy might be applicable to specific research problems within the DF domain, it would not work well for the current research question. The main limitation of this strategy is the generalization of results, especially since most criminal cases and real-world datasets are confidential in nature. Additionally, it would be difficult to replicate its results in the future.

A single quantitative research method would hold objective views, as it usually relies on numbers and proof. On the other hand, a single qualitative approach is subjective as it is interpretive in nature.

Therefore, the Mixed Methods research approach was followed in order to ensure the accuracy of the results and to get both subjective and objective views. As the central premise of the mixed methods research is that combining both quantitative and qualitative approaches would result in a better understanding of the research problem than with a mono research method.

Furthermore, the research strategies that were used included semi-structured interviews, experiments, survey, and questionnaire.

Mixed methods research as a methodology involves guiding the direction of the mixing of qualitative and quantitative approaches in the research process and the collection and analysis of data. As a method of inquiry, it involves the mixing of quantitative and qualitative data in a single or a series of studies. (Creswell & Clark, 2007, p. 5)

More specifically, the mixed approach that was followed fell under the partially integrated mixed

methods approach, where both qualitative and quantitative methods complemented each other. In order to support the interpretation and conclusions reached each set of data were collected, analyzed, and presented separately. (Saunders, et al., 2012, p. 166)

The first methodology that was used relied on a qualitative approach that used semi-structured interviews with the study participants. Glesne (2011, p. 102) explained that in a semi-structured interview the researcher would start with some questions and remain open to reforming or adding to them. The aim of these interviews was to identify the regular methods the participants had used during DF investigations, which established the need for techniques similar to those provided by Text Data Mining (TDM) and Relevance Feedback (RF) techniques in DF. Therefore, participant interviews were considered crucial.

The second methodology was based on a quantitative approach that used different experiments. The aim of these experiments was to get quantitative proof of the developed solution. The experiments were explained in details in Chapter Six: Experiments Design.

Additionally, a survey and a questionnaire were provided to each of the participants after they completed the study experiments. The survey contained close-ended questions and was used to supplement the preliminary results of the interviews and experiments. The evaluation of the experiments' results was taken into account, to study their assessment of the PoC techniques in comparison to the other DF tool they used during the experiments. The overall results were provided to the participants afterwards.

Furthermore, the questionnaire was used to measure the participant's satisfaction with their use of the PoC. The criteria used in the questionnaire included the following aspects:

1. Content usefulness to the investigation.
2. The clustering of information is appropriate.
3. Suggested documents contribute to the overall solving of the investigation needs.
4. Relevance of the suggested documents to requested user query or topic of interest.
5. Discovery of new information that relate to the investigation.
6. Ease of use.

4.2. Reliability

Reliability is an important aspect of research quality, as it shows that if the techniques of data collection and analytical procedures were repeated by another they would generate consistent results with those identified by the study. Validity is another aspect of research quality, as it shows the extent to which the used methods and research findings accurately measure what they were meant to measure. (Saunders, et al., 2012)

In general, there are many threats to reliability; for example, bias and error. Bias can be reflected in the researcher's own views or ideas, or it could be any factor that might adversely induce a false response from a participant. An Error can be any factor that might alter the researcher's interpretation or adversely alters the performance of a participant.

Therefore, the use of a mixture of methodologies makes sense in this study; as the results from the quantitative and qualitative approaches fed into the development of the techniques that were used in the experiments. Furthermore, the participants' responses provided feedback of the proposed solution. Additionally, the quantitative approach was vital to proving the validity of the proposed framework, as it relied on computational techniques.

For the second part of the quantitative approach, the participants' sample in the experiments

consisted of senior DF investigators with practical experiences in solving real world crime cases that relied on the contextual relevance of the evidence found. And as discussed previously, in order to successfully evaluate the performance of the proposed framework the selected evaluation measurements were used to assess the results.

To ensure the validity and reliability of the study before and after the experiments, those participants were interviewed by the researcher. These interviews gave a better understanding of the participants' previous experiences with traditional DF tools. Furthermore, the approval of each of the participants to be audio recorded during the interviews was requested, in order to take full advantage of their input. However, as only one participant agreed to be audio recorded during the interview, the other interviews consisted of the researcher taking notes during each interview.

All necessary approvals were obtained beforehand from the Emirates Electronic Evidence Center (EEEC) for the participation of their staff in the study. The interviews took place at the EEEEC premises in Abu Dhabi (UAE) and lasted for about an hour for each participant. To ensure the authenticity and trustworthiness of this stage of the study, the interviews' results were transcribed. Copies of each interview were provided to each participant, to ensure their approval, and that their thoughts and ideas were represented accurately.

4.3. Ethical Considerations

Protecting the confidentiality of this study was an important issue, especially that real world crime data were used in the later stages of the research. As the confidentiality of the used dataset and results were protected, no privacy issues shall arise from either the qualitative or

quantitative parts of this research. And as the researcher had access to real world data, authorization was taken prior to the beginning of the experimentation and interview stages.

A written informed consent was provided to, and was signed by, the Director of the EEEEC of Abu Dhabi Police, whose staff participated in the study. The participants were asked to take part in the study and had the right to withdraw from it at any point if they deemed it necessary.

The researcher has over twelve years experience in the fields of Information Security and Digital Forensics, nine of which had been focused in the area of digital forensics. In addition, the researcher has previous experience with conducting interviews and highly developed observation skills when it comes to case examinations.

Therefore, and in the hope of achieving better subjective results for this study (absent personal biases) the intent of conducting interviews with other experts was an attempt to gain an insight into their practical experiences; rather than relying on the researcher's own prejudice from which this study sprang forth to address the main problem.

Chapter Five: Proposed Framework

This chapter explains the proposed framework, its components, and workflow. The last section previews the Proof-of-Concept (PoC) tool.

5.1. Framework Components

This research proposes a new framework to help solve the problem of the high information retrieval overhead and to speed up processing of text files found during Digital Forensics (DF) investigations. The framework combines Clustering, Classification, TF-IDF similarity, and Elasticsearch with the purpose of making the relevance feedback cycle more efficient and to reach relevant results faster. The proposed framework is shown below in (Figure 5.1), while the framework components are explained in (Table 5.1).

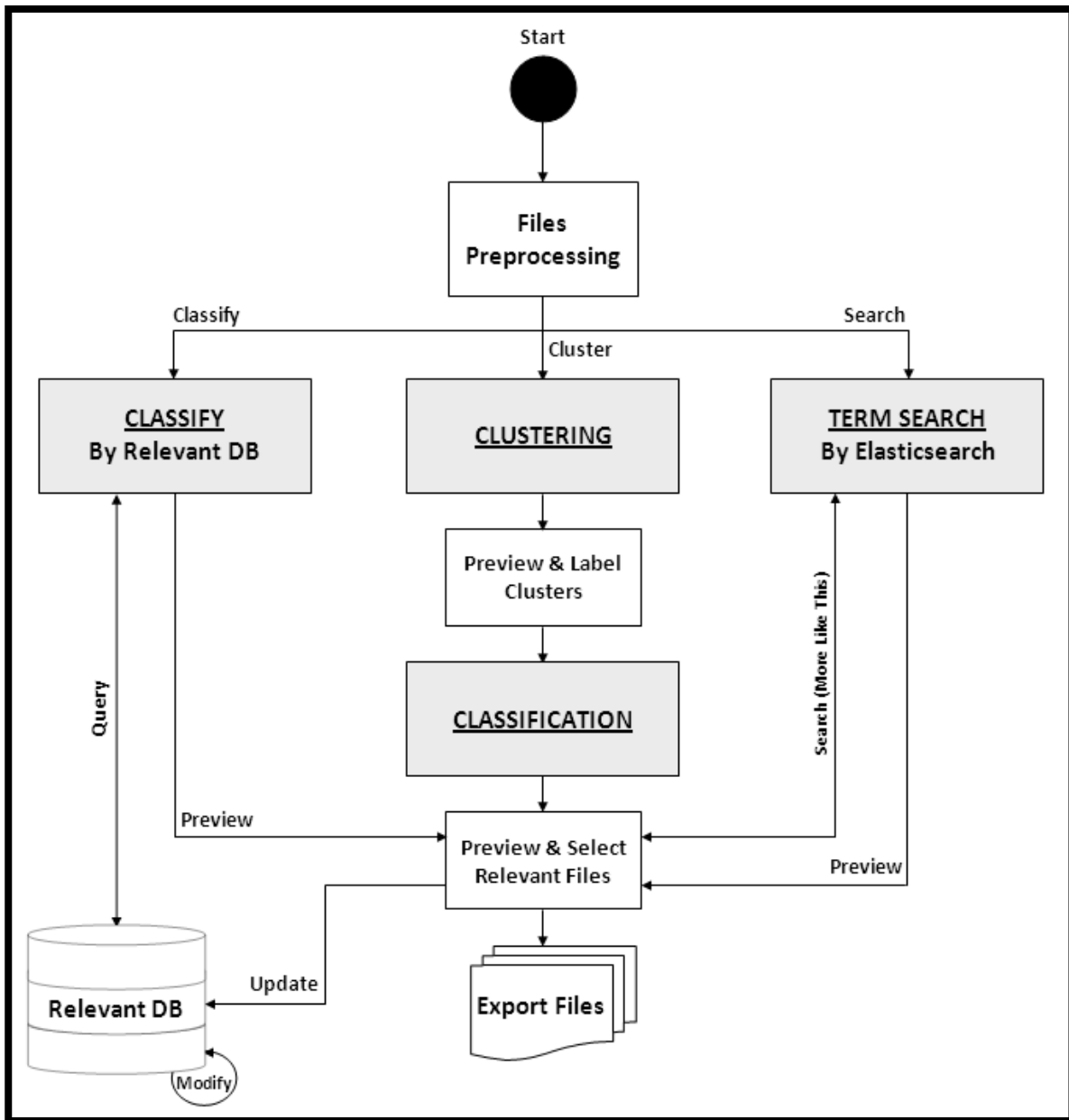


Figure 5.1: Proposed Framework

#	Process	Description
1	Preprocessing of files	Preprocessing of files might include running them through an antivirus scan to ensure they're clean and recognize any pictures containing characters using the Optical Character Recognition (OCR) feature. These features were not included in the PoC. The preprocessing of the test set files is explained in Experiments Dataset (Section 6.2).
2	Term Search by Elasticsearch	Using a term or a list of terms the user can run a search for their appearance in files.
3	Classify by Relevant DB	Using the RelevantDB the user has the option of automatically classifying a collection of files according to previously stored relevancy information in the database.
4	Clustering	Clustering of a subset of files where the user decides (estimates) the number of clusters that can be created.
5	Preview & label clusters	The user reviews the files in the clusters by previewing each file or using the Word Clouds feature that show the Top Terms that occur in each cluster. Then the user decides on what to label each cluster. The label could indicate that a file is relevant, non-relevant, crime specific label, etc.
6	Classification	The remaining files are automatically classified using the previous process based on the label that was assigned.
7	Preview & select relevant files	The user reviews and selects files that can be used to search for similar files (using the <i>More Like This</i> ⁷ function from Elasticsearch), adds them to the Relevant DB, or exports them to a specified location. The More Like This (MLT) is provided by Elasticsearch using the Lucene scoring formula ⁸ whereby the program searches for all documents similar to the given input document(s), the MLT extracts the text from the given document, analyzes it, then provides the Top Terms with the highest TF-IDF.
8	Export files	The user has the option to export selected files from the previous process to a specified location.
9	Relevant DB	RelevantDB is a MySQL database that stores selected files by the user with their classification; the records can be modified by the user later on.

Table 5.1: Proposed System Components

⁷ <https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-mlt-query.html>

⁸ https://lucene.apache.org/core/4_9_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html

5.2. Framework Workflow

In order to run the experiments of this study a simple User Interface (UI) was created with the proposed components running in the background. The UI was created using an HTML interface, while the processing of information relied on RapidMiner Server and Studio in the background. A Proof-of-Concept (PoC) tool was developed to allow for the testing of the proposed framework. The basic steps that a user can take to use the UI are explained in (Table 5.2).

Option	Description
Select Option	The user is presented with multiple options to either run a Term Search, Model & Classify, or Classify by Relevant DB.
Term Search	The user enters search terms manually or uploads a text file that contains them. The search terms will be processed and a list of files will be provided for the user's preview.
Model & Classify	The user enters the path to a subset of the files. In cases where the number of files is large, a subset of files could be selected and clustered; to create a Model that could be applied later on to the large dataset. In this step, the user enters the target path and decides (estimates) the number of clusters that would be created.
Classify by Relevant DB	The user enters Target Folder location path (overall dataset) on their local disk and classifies the dataset based on similarity with what was saved in the database.

Table 5.2: User Interface Main Sections

RapidMiner is a data mining tool that is useful for creating and developing predictive analytical models and text mining. Additionally, RapidMiner provides a useful Text Processing extension that could be found at the RapidMiner Marketplace. This extension provides helpful operators for statistical text analysis and Natural Language Processing (NLP).

The Text Processing extension was useful, as it supported several text formats such as plain text, PDF, and other data sources. It also provided handy filters⁹, all of which were necessary for preprocessing of text before running it through an algorithm for text analysis.

5.3. Search Concepts

A search feature in an application is considered a must have component in almost any customer facing application. It is considered a key component of most applications, particularly for a user who needs to sift through large volumes of text, for instance in data driven applications.

A User Interface (UI) is usually used to take a user's information needs by entering keywords, dates, document types, or other information in order to return a list of possibly relevant documents to the user's inquiry.

The beauty of a well designed search engine is the way it simplifies the job for the user, by showing the user relevant results without the complicated machinations that worked in the background. A user query uses terms and operators in a search engine. There are different query types and operators, such as keywords, Boolean operators, and regular expressions, which are useable to conduct a search task.

In order to optimize search results there are many techniques that could be employed. Usually an index created by search engines is used to match the user's search terms. This index consists of all the words in each document that's being searched and pointers to their locations in the documents.

⁹For example, filters for tokenization and Arabic Stopwords. Further details can be found in Appendix 3.

Additionally, the assignment of weight to terms in the index file is a useful method to getting better results. The most common weighting scheme is the Term Frequency-Inverse Document Frequency (TF-IDF).

The TF-IDF algorithm is based on the idea that commonly occurring terms in a large collection of documents are less important than terms that frequently occur in a document (TF) relative to the number of times those frequent terms occurred in the overall collection (IDF). (Ingersoll, et al., 2013, p. 48)

The TF-IDF uses inverse document frequency to differentiate between low value words (common) and high value words (uncommon). It also recognizes that it is more likely that a document is relevant for a specific word, the more that word appeared in it.

In contrast, a common word might appear many times in one document, which would be offset by the fact that the same word appeared in all the documents in the collection many times. TF-IDF was designed back when removing the most common words (i.e. Stopwords) from the index was standard practice, and since they were removed already the algorithm did not have to worry about an upper limit for term frequency. (Gormley & Tong, 2015)

A search performance measures quality and quantity of the produced results. Quality refers to the relevance of the retrieved results, while quantity refers to how many results are returned by the system in a given amount of time.

Nothing is more frustrating to a user of a search engine than when they enter a query and a list of vaguely relevant hits is returned, if any. The usual case when such results are returned is that the user takes on the trial and error approach by adding or changing keywords.

In between the user's need and the retrieved search results is the concept of relevance. Relevance can be defined as "the notion of how appropriate a set of results is for a user's query". (Ingersoll, et al., 2013, p. 70)

In order to judge the quality of results two evaluation metrics could be used; namely precision and recall. Precision refers to the number of relevant documents from the retrieved search results, while recall refers to the number of all retrieved documents from the dataset.

A useful technique that could be used to make decisions on the search quality is the concept of a focus group. In a focus group a number of users would be asked to use a system and would need to explicitly identify relevant and non-relevant documents. This technique is useful depending on the number of participants who can also provide feedback on a system's usability. (Ingersoll, et al., 2013, p. 71)

For the purposes of this study, the focus group approach was selected to evaluate the quality of the proposed framework. The invited participants of this study had the chance to test the proposed framework, answer questions related to the dataset, and indicate the relevance of retrieved results.

5.4. Proof-of-Concept Preview

The Proof-of-Concept (PoC) is an example of how the framework could be implemented. This section previews the PoC tool, which was used during the PoC Experiments. The user starts the process by selecting an option from the Options Menu (Figure 5.2). By performing a Term Search, the user can run a search for their appearance in files using a term or a list of terms (Figure 5.3).

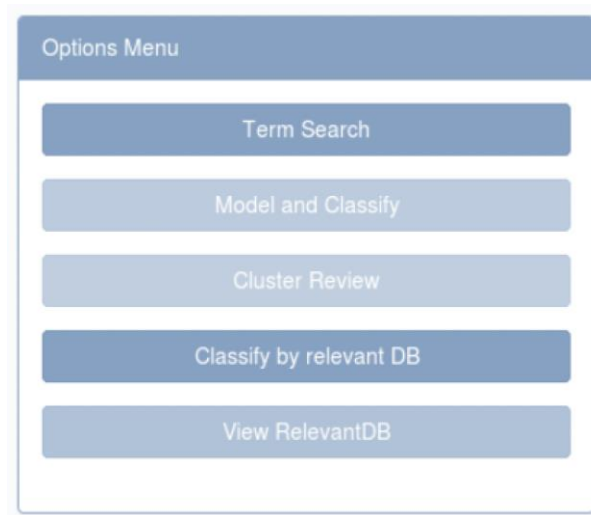


Figure 5.2: Options Menu

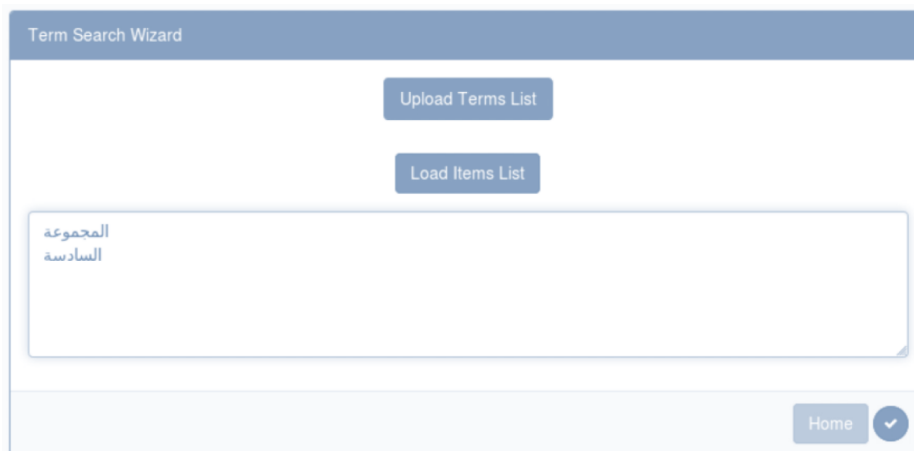


Figure 5.3: Term Search Wizard

In the next screen of running a Term Search (Figure 5.4), the user will be given the option to sort by similarity (Score), preview file contents (View File), or view the top terms within each file (Word Cloud). The user is given the chance to preview file contents (Figure 5.5) or view top terms within each file using the Word Cloud option (Figure 5.6).

WORD	File Name	View File	Word Cloud	Score
المجموعة السادسة	68.pdf	View File	Word Cloud	7.057069
المجموعة السادسة	S5.pdf	View File	Word Cloud	6.7159414
المجموعة السادسة	86.pdf	View File	Word Cloud	6.542426
المجموعة السادسة	S05.pdf	View File	Word Cloud	6.2381477
المجموعة السادسة	S06.pdf	View File	Word Cloud	4.320959
المجموعة السادسة	409.pdf	View File	Word Cloud	3.680786

Figure 5.4: Term Search (Review)

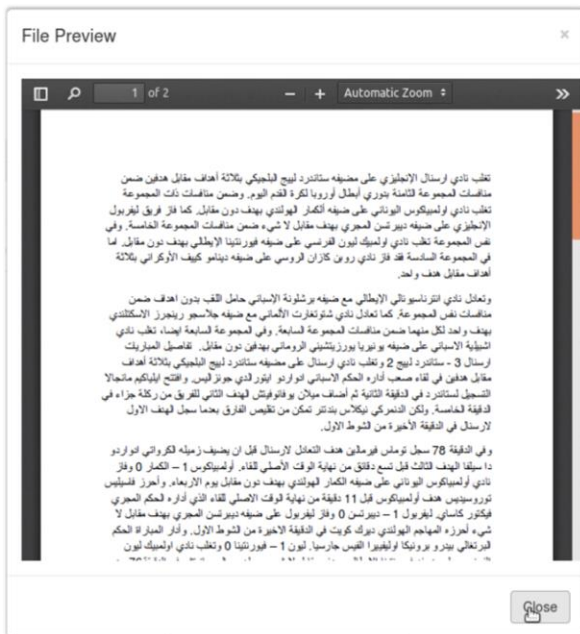


Figure 5.5: Term Search (View File)



Figure 5.6: Term Search (Word Cloud)

In the Model & Classify option (Figure 5.7), the user enters the path to a subset of the files. In cases where the number of files is large, a subset of files could be selected and clustered; to create a Model that could be applied later on to the large dataset. In this step, the user enters the target path and decides (estimates) the number of clusters that would be created.

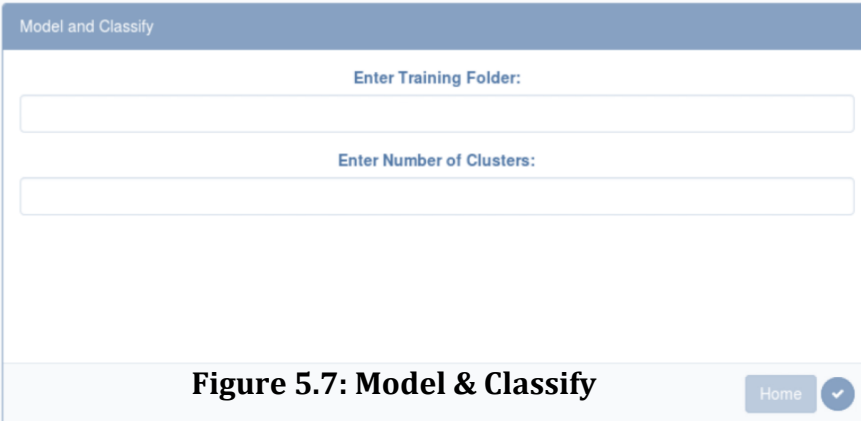


Figure 5.7: Model & Classify

The user can preview the files in each cluster to decide that cluster's label, as shown in (Figure 5.8). The label could indicate that a file is relevant, not-relevant, crime specific label, etc. Once all clusters had been labeled by the user, they enter the Target path.

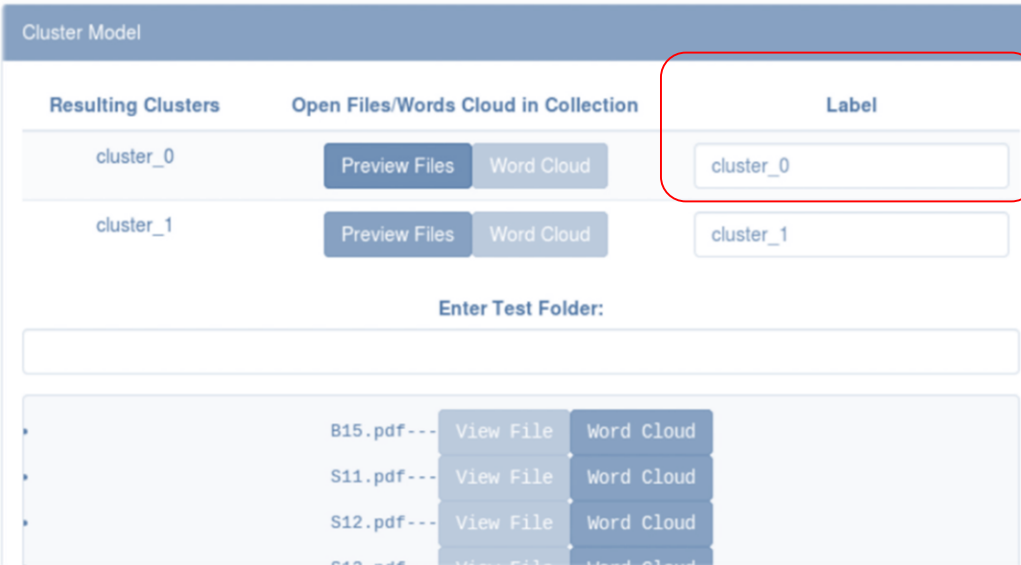


Figure 5.8: Cluster Model

User enters label for each cluster

User enters path to Target

The user is given the chance to preview the files that were automatically classified using the previous process based on the label that was assigned by the user, as shown in (Figure 5.9). The user is given different options (Figure 5.10), such as to export the selected files, use the (More Like This) feature, add selected files to Relevant DB, or Reset the whole process.

File	Predicted Label	Open File	Document Preview	Word Cloud	Checkbox
10.pdf	relevant	Download File	View File	Word Cloud	<input type="checkbox"/>
101.pdf	relevant	Download File	View File	Word Cloud	<input type="checkbox"/>
103.pdf	relevant	Download File	View File	Word Cloud	<input checked="" type="checkbox"/>
104.pdf	relevant	Download File	View File	Word Cloud	<input type="checkbox"/>

Figure 5.9: Classification Results (Review)

75.pdf	Non	Download File	View File	Word Cloud	<input type="checkbox"/>
79.pdf	relevant	Download File	View File	Word Cloud	<input checked="" type="checkbox"/>
8.pdf	Non	Download File	View File	Word Cloud	<input type="checkbox"/>

Showing 1 to 100 of 116 entries

Previous 1 2 Next

Enter Results Folder:

Home Download Selected Export Selected More Like This ! Add to DB Reset

Figure 5.10: Classification Results (Options)

The user can make a selection in the previous step and choose to use the “More Like This” feature to find similar files. In the “More Like This” option (Figure 5.11), the user will be taken to another page where they can sort the files by their similarity score and preview files either by their contents or associated Word Cloud.

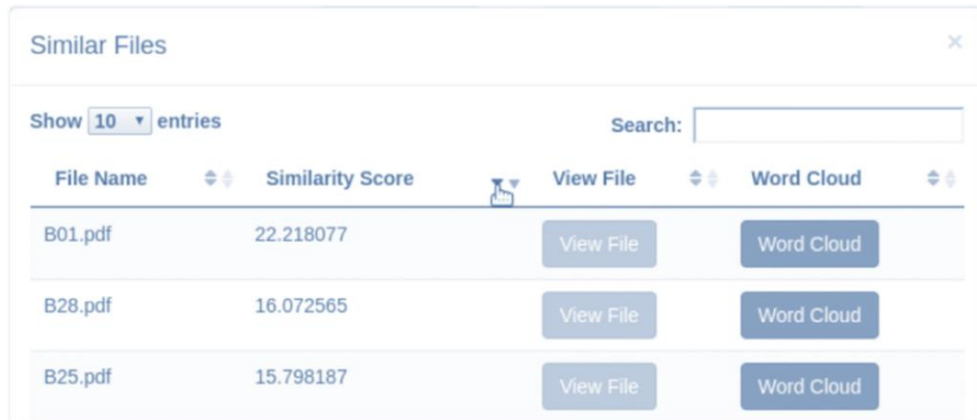


Figure 5.11: The More Like This Option

In the Classify By Relevant DB option, the user can rely on the Relevant Database to automatically classify new files (Figure 5.12). In addition, they can modify the Relevant Database (Relevant DB) records using the View Relevant DB option (Figure 5.13).



Figure 5.12: Classify by Relevant DB

View Database						
Show 10 entries		Search: <input type="text"/>				
File Name ^ ^	Category ⇅	Preview ⇅	Date Added ⇅	WordCloud ⇅	Select ⇅	
101.pdf	relevant	View File	2018-11-07T19:38:01.827Z	Word Cloud	<input type="checkbox"/>	
113.pdf	relevant	View File	2018-11-07T19:38:03.053Z	Word Cloud	<input type="checkbox"/>	
2.pdf	relevant	View File	2018-11-07T19:38:03.434Z	Word Cloud	<input type="checkbox"/>	
3.pdf	relevant	View File	2018-11-07T19:38:03.648Z	Word Cloud	<input type="checkbox"/>	
337.pdf	relevant	View File	2018-11-07T19:38:03.772Z	Word Cloud	<input type="checkbox"/>	

Figure 5.13: View RelevantDB

Chapter Six: Experiments Design

This chapter provides information on the setup of all the experiments, including the experiments goals, dataset, and the accuracy measures used to evaluate the performance.

6.1. Experiments Goals

The first part of the quantitative approach was based on experiments conducted by the researcher, while the second part ran on a number of experiments with experienced Digital Forensics (DF) investigators to analyze their use of the solution. There were four main experiments, as explained in (Table 6.1).

#	Experiment Title	Experiment Goal	Related Research Aim
1	Clustering Component Experiment	Identify the best clustering algorithm to integrate it into the proposed solution.	Research Aim (2): Propose a framework to incorporate TDM and an investigator's relevancy feedback.
2	Classification Component Experiment	Identify the best classification algorithm to integrate it into the proposed solution.	
3	Search & Visualization Experiment	Identify the best search method to interact with the clustered information generated from the proposed framework.	
4	PoC Experiment	Compare the performance quality between the proposed solution (PoC) and traditional DF tools.	Research Aim (3): Implement a Proof-of-Concept (PoC) for the proposed framework and evaluate its results.

Table 6.1: Experiments Goals & Relation to Research Aims

According to Hakim (2000), an experiment helps study the effects of a change in an independent variable causing a change in another dependent variable. The Independent Variable (IV) is the

one that causes change to a Dependent Variable (DV). The different components, i.e. Clustering or Classification algorithms, formed the Independent Variable, while the performance evaluation metrics formed the Dependent Variables.

6.2. Experiments Dataset

The goal of the three initial experiments was to identify the ideal components to use for the proposed solution; a number of tests took place using Arabic text-based datasets. Even though this work focused on Arabic text, the proposed framework is language independent. There were three main reasons for this focus. First, the Arabic language is more difficult to deal with because of its complex linguistic structure. (Farghaly & Shaalan, 2009)

Second, Arabic is predominantly used in the Middle East region, where this research originated. Third, from the professional experiences of the researcher over the past decade, text heavy analysis was almost always required when it comes to terrorism cases, which were prominently based on the formal Arabic language.

The selected datasets were split into a Training Set and a Test Set. The Training Set is typically used during the training of an algorithm, while the Test Set is used in order to evaluate its performance. The dataset used for each experiment along with the file distribution for Training and Test Sets is shown in (Table 6.2).

The experiments relied on three main datasets, two were the public datasets “bbc-arabic-utf8” and “osac-utf8”, and the last one was created by the researcher from confidential real world criminal cases. The latter of which was used for the fourth experiment with the participants. The reason for using two different public Arabic datasets was to avoid overfitting. In addition, Arabic

News articles are typically written in the formal Arabic language.

#	Experiment Title	Dataset Used	Test	Training Set (Files)	Test Set (Files)
1	Clustering Component Experiment	1. bbc-arabic-utf8 2. osac-utf8	Tested Algorithms: 1. K-Means 2. Agglomerative Hierarchy 3. Random	-	400
2	Classification Component Experiment	1. bbc-arabic-utf8 2. osac-utf8	Tested Algorithms: 1. kNN 2. Naïve Bayes 3. SVM	400	400
3	Search & Visualization Experiment	bbc-arabic-utf8	Test 1: Keyword Search (dtSearch) Test 2: Clustering & Word Cloud Website	60	60
4	PoC Experiment	1. bbc-arabic-utf8 2. Dataset created from real criminal cases	Test 1: Encase/dtSearch Test 2: PoC	72	144

Table 6.2: Experiments & Datasets

The first public dataset “bbc-arabic-utf8” was published by Saad (2010). It originally contained an Arabic corpus collected from the BBC Arabic website (bbcarabic.com) and included 4,763 text documents of different categories. The second public dataset “osac-utf8” was published by Saad & Ashour (2010). It originally contained an Arabic corpus collected from multiple websites and included 22,429 text documents of different categories.

The rationale behind using these particular datasets was the simple format of the files and because they worked well within the context of a digital forensics investigation. For instance, normal users of digital devices might read news articles or documents online, which will be seen as a normal activity for a user. By contrast, a terrorist might also look at other online text that

might help in their criminal activities. To a digital forensics investigator the ability of grouping such text based on their content is invaluable.

The Clustering Component Experiment used 400 files from the two public datasets, which were randomly selected from 5 main categories (Sports, Science & Technology, Business & Economy, Cooking, and Health). The reason for selecting 5 categories only in these experiments was to showcase the capability of the tested algorithms when it came to distinguishing similar and dissimilar files based on the category they might fall within.

The Classification Component Experiment used 800 files from the public datasets, which were randomly selected from 5 main categories (Sports, Science & Technology, Business & Economy, Cooking, and Health). The Training and Test Sets contained 400 files each.

The Search & Visualization Experiment's technical purpose was to test a text-rich sample, so news articles proved to be the most useful. It used 120 files from the "bbc-arabic-utf8" dataset, which were randomly selected from 3 main categories (Sports, Science & Technology, and Business & Economy). Furthermore, 60 of those were designated as the Training Set (20 files from each category), and 60 files were allocated for the Test Set (20 files from each category).

The PoC Experiment used a dataset that was created from select files taken from two datasets, the "bbc-arabic-utf8" and a dataset created from criminal cases. The criminal cases' dataset was created by the researcher using real criminal cases that contained confidential information on different terrorism related topics. The total number of files in the used dataset was 216 files, which were split into 72 files in the Training Set and 144 files in the Test Set. The Training Set was used by the researcher during the development of the PoC, whereas the Test Set was used by

the participants during their tests.

Additionally, no files from the Training Set that were used during the development stage of the proposed solution were reused later. To put it another way, the Test Set files were never used on the PoC prior to the participants' tests.

There were two types of files used in the overall dataset of this experiment; Good files taken from the public "bbc-arabic-utf8" dataset, and Bad files taken from the criminal cases dataset. The topics for the used Bad files ranged from weapons, to the making of explosives and toxins, and terrorism manuals and manifestos, all of which were in formal Arabic. The Good files included Arabic news articles. The distribution of files is shown in (Table 6.3).

#	Used By	Dataset Used	File Count (Total)	Good Files		Bad Files	
1	Researcher	Training Set	72	32		40	
2	Participants	Test Set	144	Group 1	Group 2	Group 1	Group 2
				32	33	40	39

Table 6.3: Training & Test Sets File Distribution

The Training Set had 32 files which were Good and 40 files that were Bad. On the other hand, the Test Set contained 2 groups of files, Group 1 and Group 2, which was comprised of 72 files each. Group 1 contained 32 Good files and 40 Bad files, while Group 2 contained 33 Good files and 39 Bad files.

Data preprocessing was done on all of the dataset files. In order to remove any identifying headlines or references, all the files were processed to remove the header information and only the body of the text remained.

DF investigators use different techniques to figure out relevant files in a case. For example, timeline analysis or file signature analysis. The PoC itself does not yet use file metadata information. Thus, in order for the participants of the PoC Experiment to focus on using the developed tool's techniques a number of measures were taken, as follows:

1. Changed all the timestamps of the files in the Test Set (i.e. Creation, Access, and Modification timestamps) to the exact date and time of all the files, thereby eliminating the examiner's ability to use Timeline Analysis.
2. Changed all the filenames into pseudorandom numbers, so the users could not use the filenames to figure out the given test questions and had to rely on the search features provided by the tested tools.
3. Normalized the file content lengths, so they would be of comparable sizes and their sizes were not a factor that gave away the answers. Additionally, this step helped prevent any effects text sizes might have on the clustering, classification, or the TF-IDF techniques.
4. The participants were given the same list of questions, but the answers as well as the filenames in the datasets were changed, so that the participants could not share the answers. They were also asked to provide the answer to each question along with the filename they have found the answer in.

6.3. Evaluation Measurements

The initial evaluation measurements that were selected took after those used traditionally within the Information Retrieval domain. Two of the most popular measures for characterizing the performance of a technique are Precision and Recall.

Precision was defined by Manning & Schutze (1999, p. 268) as a “measure of the proportion of selected items that the system got right.” Recall was defined as the “proportion of the target items that the system selected.” (Manning & Schutze, 1999, p. 269)

The F-measure is another measure that calculates the overall performance by combining precision and recall. Thus, Precision, Recall, and F-measure were used during the quantitative phase of this research for the purpose of evaluating the performance of the proposed solution and its components. The measurements that were used include the following:

1. **True Positive (TP) Rate:** the total number of items selected as relevant. The higher this rating is the better; ideally this would be (100%). This measure indicates how effective the approach is in detecting all relevant items correctly.
2. **True Negative (TN) Rate:** the number of non-relevant items correctly omitted.
3. **False Positive (FP) Rate:** the number of items incorrectly selected as relevant. The lower this rating is the better; ideally this would be (0%). This measure indicates how many items were incorrectly detected as relevant, while they were not.
4. **False Negative (FN) Rate:** the number of items incorrectly selected as non-relevant. The lower this rating is the better; ideally this would be (0%). This measure indicates how many relevant items slip undetected.
5. **Precision:** the total number of TP divided by the combined TP and FP.

$$\frac{TP}{TP + FP}$$

6. **Recall:** the total number of TP divided by the combined TP and FN.

$$\frac{TP}{TP + FN}$$

7. **F-Measure:** the overall performance is measured by combining Precision and Recall using the following calculation:

$$\mathbf{F\text{-Measure} = \frac{2 * \mathbf{Precision} * \mathbf{Recall}}{\mathbf{Precision} + \mathbf{Recall}}}$$

8. **Error Rate:** the percentage of incorrectly processed items, using the following calculation:

$$\mathbf{Error\ Rate = \frac{FP + FN}{Total\ Number\ of\ Items}}$$

9. **Accuracy:** if the error rate equals the percentage of incorrectly processed items and accuracy is the percentage of correctly processed ones, then the accuracy is calculated as follows:

$$\mathbf{Accuracy = 1 - Error\ Rate}$$

Chapter Seven: Proposed Framework Components Testing

This chapter provides information on the experiments that were carried out, explains the test methods for each of these experiments, and provides details on the participants' experiments with the proposed solution.

7.1. Clustering Component Experiment

The goal of the following tests was to identify the best clustering algorithm to integrate into the proposed framework. Rapidminer Studio (v. 7.4) was used to conduct all the tests of this experiment. In order to choose the best clustering algorithm, different algorithms were tested and compared as shown in this section.

The tested algorithms included K-Means, Agglomerative Hierarchical Clustering, and Random Clustering. K-Means and Hierarchical Clustering had been discussed previously¹⁰ in Chapter Three. In Random Clustering a random flat clustering operation is performed on the dataset. This algorithm does not guarantee that all clusters will be non-empty, and it randomly assigns examples to clusters.

The tested Agglomerative Hierarchical Clustering was using the Complete-Link strategy. The complete-link clustering uses the maximum of object distances to create its grouping.

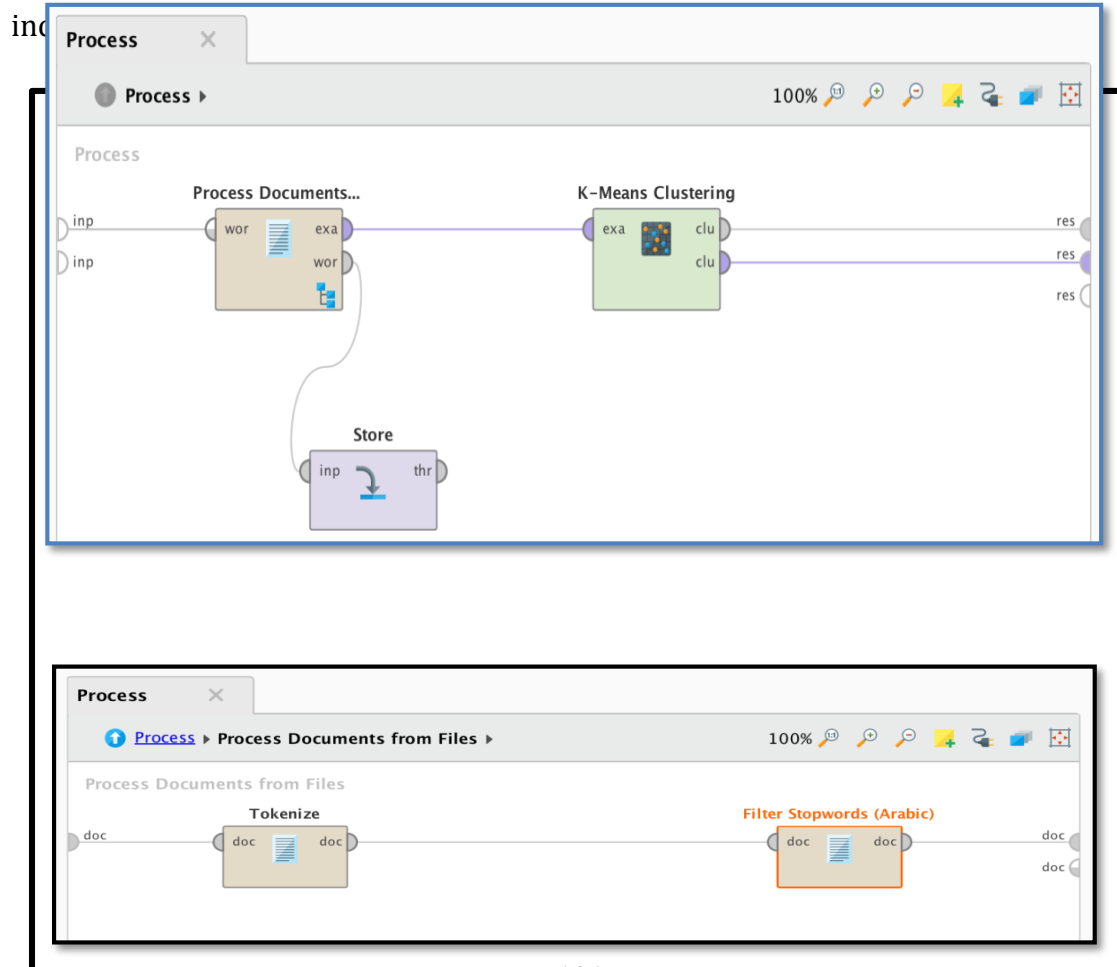
(RapidMiner, 2018)

The tests of this experiment were conducted by the researcher only. The testing method relied on

¹⁰ Further details can be found in Sections 3.1.1 and 3.1.2.

using RapidMiner, where the following took place on the dataset:

1. Each document was tokenized and the Arabic Stopwords were filtered out. The schema for the word vector creation was set to Term Frequency–Inverse Document Frequency (TF-IDF). (McGuigan, 2010)
2. Those files were then processed using the tested clustering algorithm in order to group files into different clusters, and the value of K was set to 5, as shown in (Figure 7.1).
3. The resulting clusters were manually checked by the researcher to determine the accuracy of the file distribution over the 5 clusters.
4. Based on the majority of the items and the category they belonged to, the 5 clusters were simply noted as belonging to one of the five categories, namely Sports, Technology & Science, Business & Economy, Cooking, or Health.
5. Additionally, to compare the clustering results of each of the algorithms, the (TP/TN/FP/FN) detection rates were used to calculate the evaluation measurements



7.2. Classification Component Experiment

Figure 7.1: Document Clustering Process

The goal of the following tests was to identify the best classification algorithm to integrate into the proposed framework. Rapidminer Studio (v. 7.4) was used to conduct all the tests of this experiment. The tested algorithms included the K-Nearest Neighbor (kNN), Naïve Bayes, and Support Vector Machine (SVM), which had been discussed previously¹¹ in Chapter Three.

The kNN algorithm is a simple and effective algorithm that is known to perform well in classification jobs of documents of different categories. It compares an unknown dataset with the K training dataset, based on the nearest neighbors of the unknown Example. In order to calculate the distance between the unknown Example and the training Examples different metrics can be used, including the Euclidean distance measure. (RapidMiner, 2018)

The Naïve Bayes relies on a probability model that uses Gaussian probability densities to model the Attribute data. The fundamental assumption is that given the value of the class, the value of any Attribute is independent of the value of any other Attribute. The SVM algorithm takes a set of input training data; each example is marked as belonging to one of the two categories. A model is built that assigns new examples into one of the specified categories. (RapidMiner, 2018)

The tests of this experiment were conducted by the researcher only. The testing method relied on using RapidMiner, where the following took place on the dataset:

¹¹ Further details can be found in Section 3.2: Classification.

1. The Training Set contained 400 files that were divided manually by the researcher into 5 categories. The RapidMiner classification process is shown in (Figure 7.2).
2. Each document was tokenized and the Arabic Stopwords were filtered out. The schema for the word vector creation was set to Term Frequency–Inverse Document Frequency (TF-IDF). (McGuigan, 2010)
3. Those files were then processed using the tested algorithms in order to create a Model that would be used in the next step, and Cross Validation¹² with 10-folds was used, as shown in (Figures 7.3).
4. The Model was then used to run predictions on the Test Set (Figures 7.4), which contained 400 files, and those were manually checked for prediction accuracy.
5. The classification results of each of the algorithms were compared and the (TP/TN/FP/FN) rates were used to calculate the evaluation measurements including Precision, Recall, and F-measure.

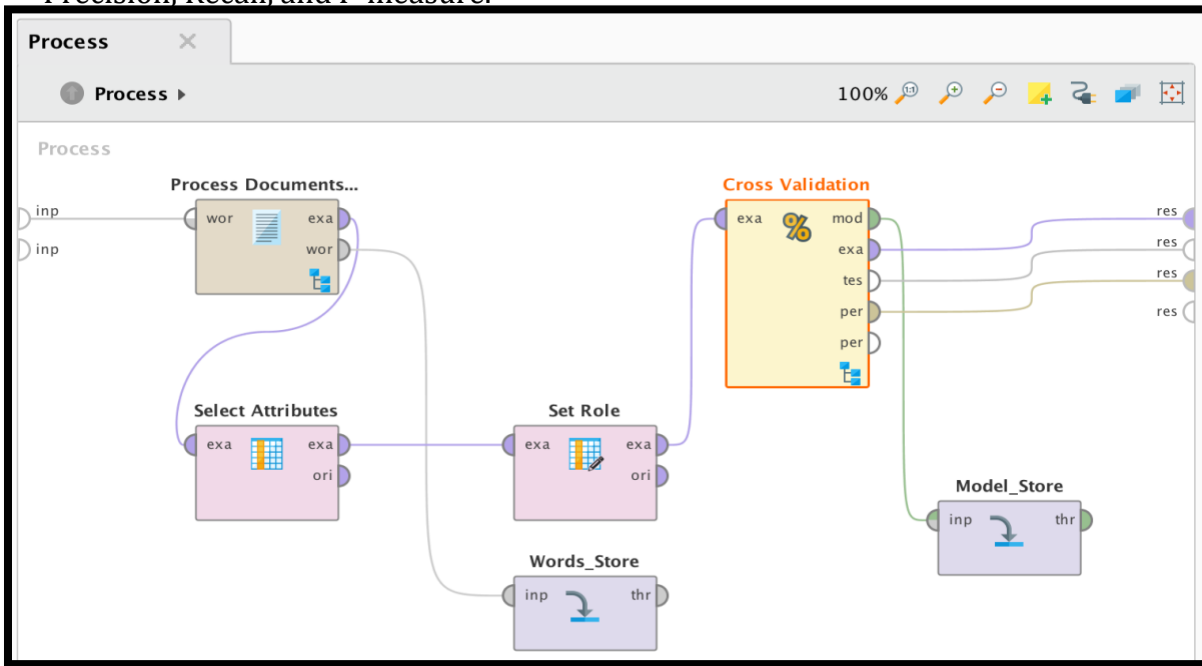


Figure 7.2: Document Processing with k-NN

¹² Further details can be found in Section 3.2: Classification.

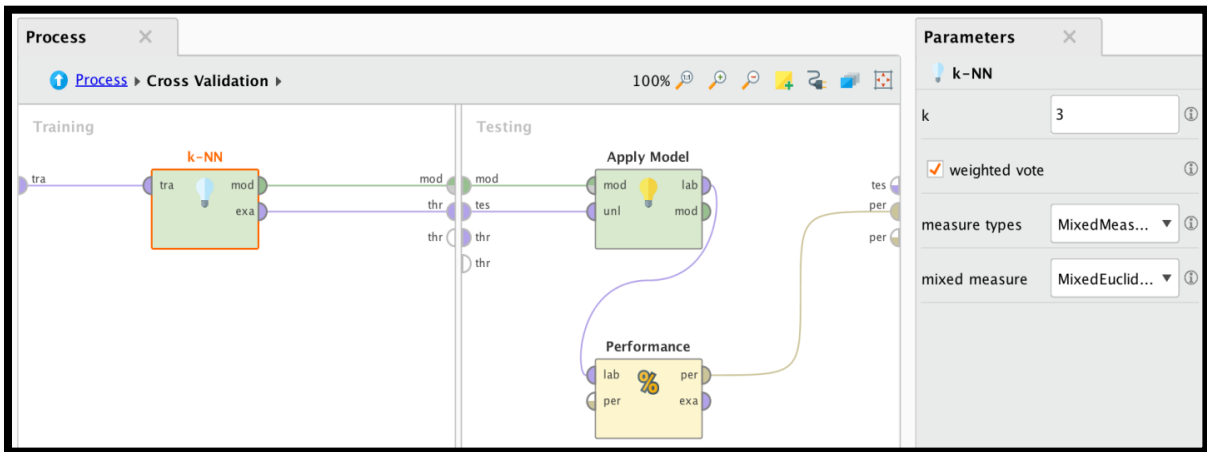


Figure 7.3: Document Processing with kNN (Cross Validation & Applying the Model)

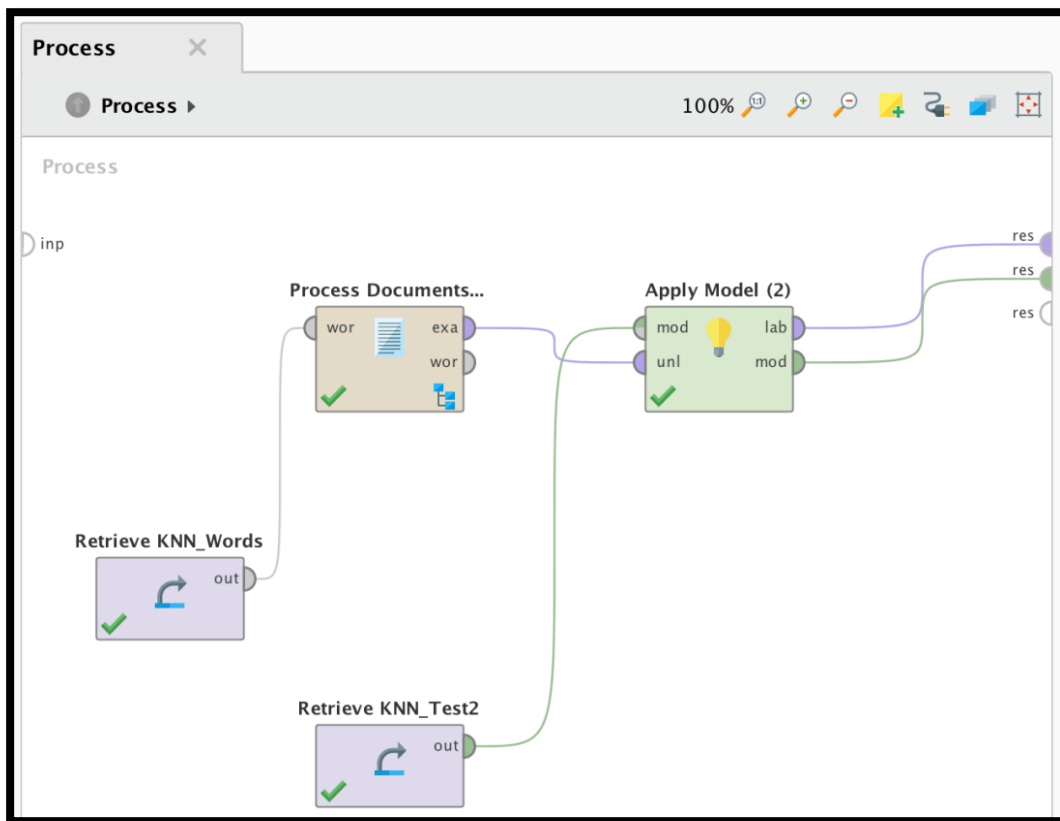


Figure 7.4: Applying the Learnt Model on the Test Set

7.3. Search & Visualization Experiment

An investigator in a Digital Forensics (DF) case uses many techniques and tools to reach useful information that relates to their investigation. A Keyword search is a useful technique when it comes to investigations where specific terms that relate to a case are known, for example specific names, locations, or numbers.

However, search terms used in normal criminal cases tend to be more generic and the DF investigator ends up having to dig deeper into the collected information to reach relevant information. Thus, many techniques were proposed and developed by the research community to assist with regards to limiting the amount of data an investigator is required to sift through.

The goal of the following tests was to identify the best search method to interact with the clustered information that would be generated by the proposed framework. The aim was to better understand how a user may formulate their search given specific questions that pertain to a dataset, and which of those methods might lead to accurate results faster. The comparison was between a search technique that relied on a Keyword Search and another that combined text-clustering techniques with Word Clouds during a search.

The tests of this experiment were conducted by participant 1. Two tests were used as the research method for this experiment, and a number of tools were used to set them up. The challenging part was determining the manner by which the two techniques in question, keyword searches and text-clustering with visualization assisted searches, could realistically be compared without the researcher's bias. Thus, two tests were designed to be carried out by a participant in the study, and their timing and accuracy were measured by the researcher.

In order to measure the effectiveness of each technique, the two tests included four questions. Three questions pertained to specific information from the articles in the dataset, while the fourth was to estimate the number of categories within the dataset.

The participant had to search the test set first using the keyword search as their main method, and in the second test had to search the clustered files by relying on the prepared Word Clouds web pages.

The word grouping used during the second test, was basically a collage of words with emphasis on the top frequent terms taken from the clustered files. To that end, the user had to look for the answers by selecting clusters based on those words. Then, they had to navigate between the clusters until they reached a specific group of similar documents. Finally, they had to manually search their contents to reach their target.

7.3.1. Keyword Search Test

For the first test¹³, the files in the dataset were indexed using dtSearch Desktop¹⁴ (version 7.83). The tool was used to index the files from the Test Set using the dtSearch Indexer, whereby it stored the location of each word from the collection to be easily searched using the dtSearch Desktop. The indexed results were searchable by the user querying the collection for specific terms, which could be a simple word, Boolean searches, or regular expressions.

The user was then provided with a list of all retrieved results generated from their search term, and shown in the lower part of the screen a preview of their selected item from that list. While

¹³ Further details can be found in Appendix 2.1: Setup with dtSearch.

¹⁴ The tool is available on: www.dtsearch.com

conducting this test, the participant was provided with an overview of how the tool works and asked to answer the test questions using the Test Set.

The participant was then asked to write down each answer and from which file they found the answer to each of those questions. Furthermore, during the test each question was timed by the researcher to be used as data during the analysis of results at the end.

7.3.2. Visualization Assisted Search Test

For the second test¹⁵, the same files in the dataset were clustered using RapidMiner Studio¹⁶ (version 7.1). A number of processes were used to prepare the documents. For example, each document was tokenized and the Arabic Stopwords were filtered out. The schema for the word vector creation was set to Term Frequency–Inverse Document Frequency (TF-IDF). (McGuigan, 2010)

Those files were then clustered using the K-Means and Euclidean Distance algorithms in order to group similar and dissimilar files into different clusters. The value of K was set to 6, based on the results achieved during the tests that ran on the Training Set, which was the number of clusters that would be formed from the data. In order to provide the participant with a user friendly interface to navigate the generated word clouds, a simple website¹⁷ was created and presented to the user (Figure 7.5).

¹⁵ Further details can be found in Appendix 2.2: Setup with RapidMiner.

¹⁶ The tool is available on: www.rapidminer.com

¹⁷ Further details can be found in Appendix 2.2.2: Website Design.

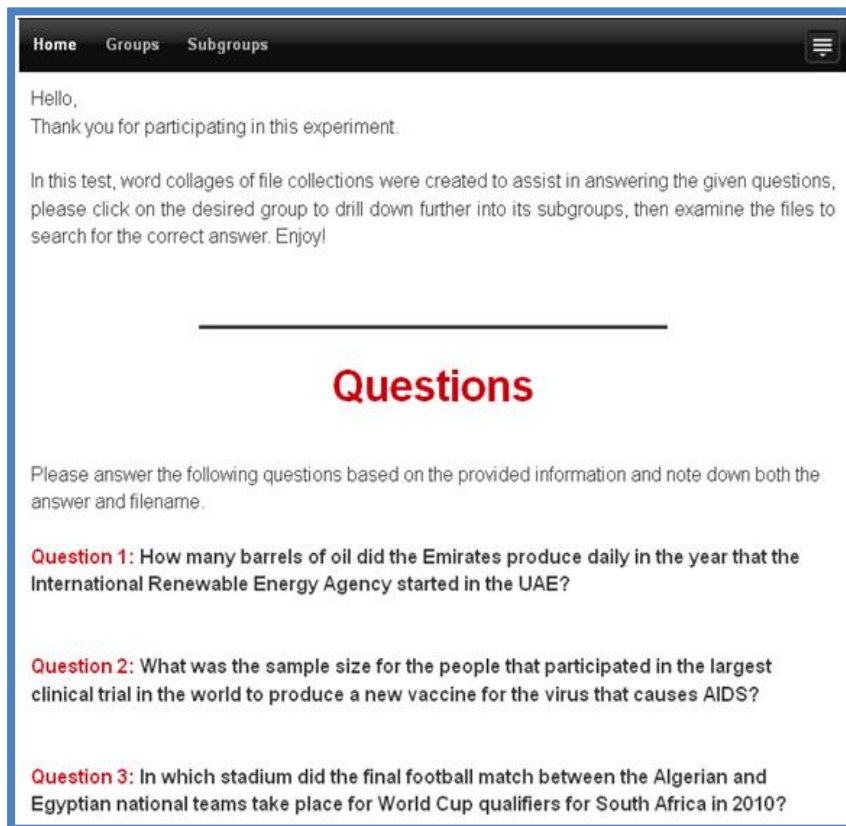


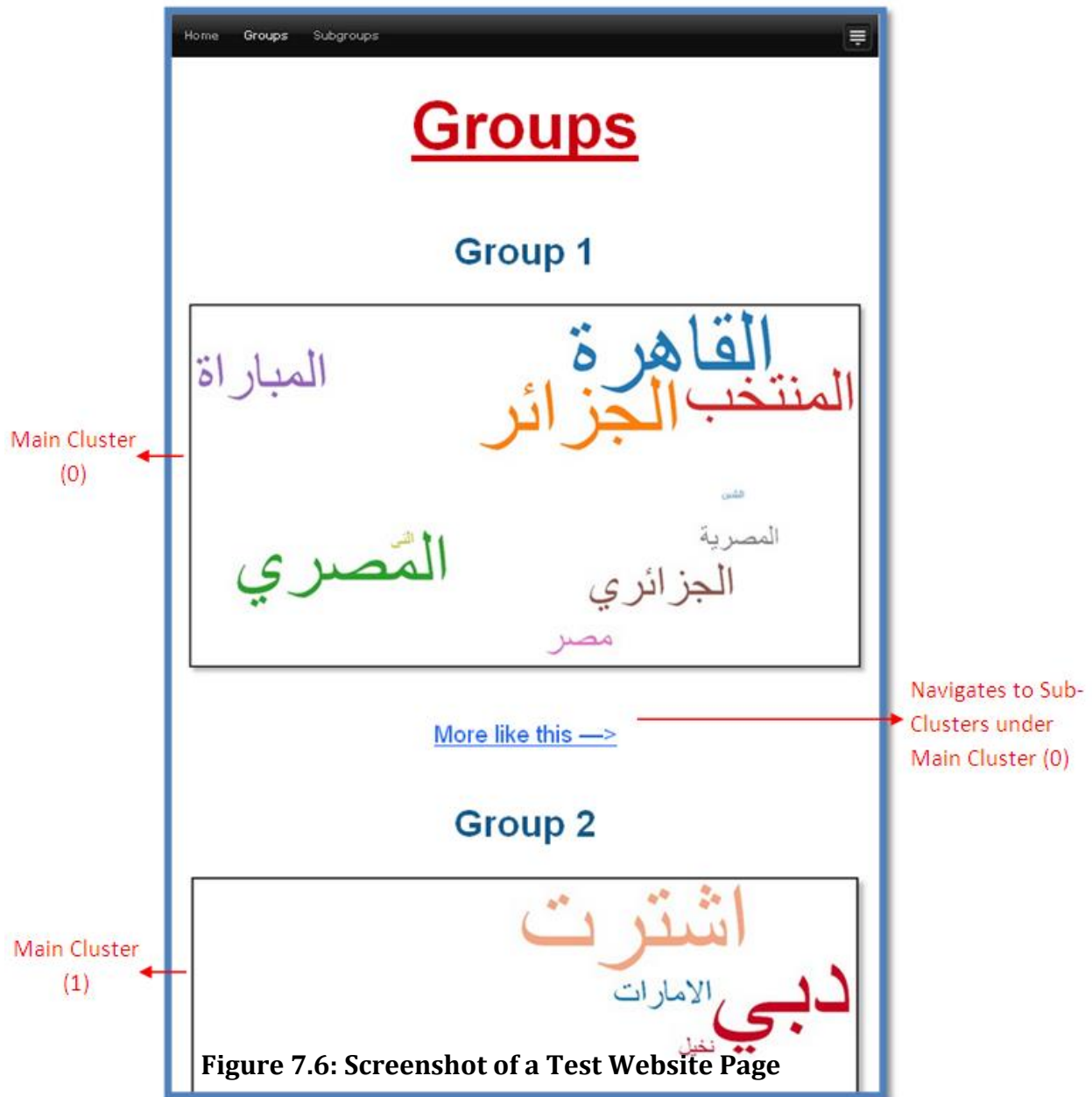
Figure 7.5: Screenshots of Test Website Pages (Main Page)

Based on the Word Clouds from the collection, article categories can easily be determined. The Test Set contained three main categories; namely, Sports, Business & Economy, and Science & Technology. The category distribution over the Main Clusters is shown in (Table 7.1). This information was not shared with the participant.

Main Cluster	Category
0	Sports
1	Business & Economy
2	Science & Technology
3	Sports
4	Business & Economy
5	Science & Technology

Table 7.1: Cluster Category

The participant was provided with an overview of how to navigate the website and move from the Main Clusters to their Sub-Clusters, and how to preview files from each cluster (Figure 7.6). Furthermore, they were given the test questions and were asked to note down each answer and filename. Finally, during the tests each question was timed by the researcher.



Word Clouds were created from the Main and Sub-Clusters¹⁸ centroid's top 10 terms, shown in (Table 7.2). All the Word Clouds were presented to the participant in a simple website.

Table 7.2: Main Clusters Word Clouds & Sub-Clusters Files			
Main Cluster #	Sub Cluster #	Filename (.txt)	Main Cluster (Word Cloud)
0	0	30	
		48	
		52	
		58	
	1	13	
		17	
		35	
	2	26	
		37	
		49	
50			
		53	
1	0	3	
		4	
		15	
		33	
	1	20	
	2	27	
29			
2	0	1	
		2	
		9	
		32	
	1	16	
		23	
		54	
	2	10	
		41	

¹⁸ Word Clouds for each of the Sub-Clusters can be found in Appendix 2.

Table 7.2: Main Clusters Word Clouds & Sub-Clusters Files			
Main Cluster #	Sub Cluster #	Filename (.txt)	Main Cluster (Word Cloud)
3	0	44	
		51	
		57	
	1	59	
	2	22	
		31	
		56	
60			
4	0	12	
		14	
		42	
	1	24	
		28	
		40	
		43	
		6	
	2	11	
		45	
		46	
		55	
5	0	8	
		18	
		34	
	1	19	
		25	
	2	5	
		7	
		21	
		36	
		38	
39			
47			

Table 7.2: Main Clusters Word Clouds & Sub-Clusters Files

7.4. PoC Experiment

This section presents the testing methods used during the PoC Experiment and explains how the tests were conducted by each of the participants. The purpose of this experiment was to compare the performance quality between the proposed solution, using the PoC, and traditional Digital Forensics (DF) techniques.

The tests of this experiment were conducted by the participants. The proposed solution was compared to the traditional search techniques DF investigators typically use during their examinations. In general, the tests were conducted as follows:

1. Given a dataset the participant was asked to use their preferred DF tool to locate information relevant to the criteria of interest; for example, to answer specific questions related to certain files. These were the Keyword Search related questions.
2. The participant was asked to use their preferred DF tool to identify the relevant documents to a given criminal case scenario. This was the Relevancy related question.
3. The participant was then asked to use the proposed solution's Proof-of-Concept (PoC) tool to locate information relevant to the criteria of interest; for example, to answer specific questions related to certain files. These were the Keyword Search related questions.
4. The participant was asked to use the PoC tool to identify the relevant documents to the given criminal case scenario. This was the Relevancy related question.
5. Evaluation of relevancy of retrieved documents was later measured by the researcher, taking into account the selected Evaluation Measurements (Section 6.3), and the time spent on each task.

In order to remove any doubts of the proposed framework's PoC tool being overfitted for the datasets, the experiment Test Set was not used during the development Training stage of the PoC tool by the researcher. The Test Set was only used by the participants during their tests. Because

the classifier needed to be tested experimentally on different datasets, to insure that the classifier could achieve acceptable results.

Avoiding overfitting the framework to a specific pattern of data was important. As a result, the tests that ran during the experiments were basically a proxy for the PoC tool’s performance on unknown future data. This was proven by the use of the Test Set by the participants, as the PoC tool did not previously handle such data.

The experiment tests were setup as follows:

1. There were two tests for each participant (Test 1 and Test 2).
2. Each Participant was provided with a folder for each test, as shown in (Table 7.3), which contained (72) files in each folder.

Note: There were two file groups (Group 1 and Group 2) which were switched between the participants to show the consistency of the test results of each tool regardless of the dataset used.

	Test 1 (Traditional DF Tool Test)	Test 2 (PoC Test)
Participant 1	Group 1	Group 2
Participant 2	Group 2	Group 1
Participant 3	Group 1	Group 2

Table 7.3: Groups Distribution over Tests

3. In Test 1, each participant used their preferred DF tool to answer the test questions.
4. In Test 2, each participant used the developed PoC tool to answer the test questions.

Note: Even though the same questions were used for the two tests, both the answer and the filename were different in the file groups.

5. The results were averaged to provide a better picture for the performance of all the participants with each tool.

The test with Participant 1 was conducted as follows:

1. In Test 1, Participant 1 was given Group 1, which included (72) files.
2. Participant 1 was asked to answer Test 1 questions.
3. In Test 2, Participant 1 was given Group 2, which included (72) files.
4. Participant 1 was asked to answer Test 2 questions.

The test with Participant 2 was conducted as follows:

1. In Test 1, Participant 2 was given Group 2, which included (72) files.
2. Participant 2 was asked to answer Test 1 questions.
3. In Test 2, Participant 2 was given Group 1, which included (72) files.
4. Participant 2 was asked to answer Test 2 questions.

The test with Participant 3 was conducted as follows:

1. In Test 1, Participant 3 was given Group 1, which included (72) files.
2. Participant 3 was asked to answer Test 1 questions.
3. In Test 2, Participant 3 was given Group 2, which included (72) files.
4. Participant 3 was asked to answer Test 2 questions.

The researcher observed the time spent by the participants to answer each question and complete all the tasks, without interfering. Finally, the researcher manually analyzed the accuracy of the results for all the tests from all the participants, and then averaged the complete results to compare the overall performance.

Chapter Eight: Evaluation of Results

This chapter presents the results¹⁹ of all the experiments that were outlined in the previous chapter. The last section shares excerpts from the transcribed interviews and questionnaire results.

8.1. Clustering Component Experiment Results

The results of the Clustering Component Experiment are shown in (Table 8.1). The K-Means algorithm outperformed the other two with an average accuracy of 97.1%. Based on these results, the K-Means algorithm was selected for the proposed framework.

Measures	Clustering Algorithms														
	K-Means					Agglomerative Hierarchical					Random Clustering				
Precision	80%	100%	100%	100%	100%	95%	100%	100%	98%	56%	58%	93%	100%	100%	63%
Recall	100%	100%	100%	92%	58%	97%	55%	100%	93%	100%	99%	68%	100%	26%	88%
F-Measure	89%	100%	100%	96%	73%	96%	71%	100%	96%	72%	74%	78%	100%	41%	73%
Error Rate	7%	0%	0%	2%	5%	3%	9%	0%	2%	10%	21%	8%	0%	20%	8%
Error Rate Avg.	2.90%					4.80%					11.10%				
Avg. Accuracy	97.10%					95.20%					88.90%				

Table 8.1: Clustering Comparison Results

8.2. Classification Component Experiment Results

The results of the Classification Component Experiment are shown in (Tables 8.2 – 8.3). Even though the results of kNN and Naïve Bayes algorithms were closely matched, the kNN algorithm was selected for the proposed framework. This was based on the average results between the

¹⁹ The discussion of the results can be found in Chapter Nine.

Training and Test stages of these two algorithms, as the average accuracy of kNN was 97.35%, as shown in (Table 8.4).

Classification Algorithms															
Measures	kNN					Naïve Bayes					SVM				
Precision	97%	74%	100%	100%	100%	97%	75%	100%	98%	96%	98%	69%	100%	96%	94%
Recall	97%	74%	100%	100%	100%	97%	75%	100%	98%	98%	98%	69%	100%	96%	96%
F-Measure	97%	74%	100%	100%	100%	97%	75%	100%	98%	97%	98%	69%	100%	96%	95%
Error Rate	2%	11%	0%	0%	0%	2%	10%	0%	1%	1%	1%	13%	0%	2%	1%
Error Rate Avg.	2.50%					2.75%					3.36%				
Avg. Accuracy	97.50%					97.25%					96.64%				

Table 8.2: Classification Algorithms Results – Training Stage (Using Training Set)

Classification Algorithms															
Measures	kNN					Naïve Bayes					SVM				
Precision	92%	93%	100%	92%	90%	92%	89%	100%	98%	87%	81%	85%	100%	100%	100%
Recall	96%	86%	100%	93%	92%	97%	79%	100%	96%	96%	99%	70%	100%	96%	84%
F-Measure	94%	90%	100%	93%	91%	94%	83%	100%	97%	91%	89%	77%	100%	98%	91%
Error Rate	4%	4%	0%	4%	2%	4%	6%	0%	2%	2%	7%	9%	0%	1%	2%
Error Rate Avg.	2.80%					2.80%					3.70%				
Avg. Accuracy	97.20%					97.20%					96.30%				

Table 8.3: Classification Algorithms Results – Test Stage (Using Test Set)

	kNN	NB	SVM
Error Rate Avg. (Training & Test)	2.65%	2.78%	3.53%
Avg. Accuracy (Training & Test)	97.35%	97.23%	96.47%

Table 8.4: Classification Algorithms Results (Training & Test Stage Averages)

8.3. Search & Visualization Experiment Results

The results between the two tested techniques were compared and presented in this section. It is worth noting that the time each tool, dtSearch and RapidMiner, spent during the process execution was 1 second for each case (Figures 8.1 – 8.2). The reason for this could be attributed to the low number of documents in the sets. In a set with a higher number of documents, the process execution times might vary and shall be taken into consideration.

Documents indexed	
Total words in index	12,437
Total documents in index	60
Documents indexed	60
Bytes indexed	465 kb
Documents removed	0

Figure 8.1: dtSearch Processed Documents Report

✓	Training Set 2.7 (2 results. Process results) Completed: Aug 9, 2016 12:31:18 PM (execution time: 1 s)
✓	Training Set 2.7_Clusters (2 results. Process results) Completed: Aug 9, 2016 1:16:06 PM (execution time: 0 s)

Figure 8.2: RapidMiner Execution Time (Main & Sub Clusters)

During the experiments, the participant was able to find the correct answers in both experiments. The three questions for the tests were different; and included questions from different categories.

The total time spent in answering the three main questions during Test 1 was (8.9) minutes, while the participant spent less than half that time (4.01) minutes to complete them in Test 2.

The purpose of the fourth question was to determine the ease by which someone could figure out how many categories were present in the sample documents.

The participant concluded that there were 14 different categories distributed over the different articles in Test 1, and 7 different categories in Test 2. In the first test, the overall time spent to complete all the questions was (18.3) minutes, while in the second test (4.4) minutes. The detailed timings recorded during the tests are shown in (Table 8.5).

Test Question #	Test 1 Timing (Minutes)	Test 2 Timing (Minutes)
Q1	5.23	1.42
Q2	2.47	2.17
Q3	1.2	0.42
Q4	9.31	0.39
Total	18.3	4.4

Table 8.5: Experiment Timing

The overall timed performance for the participant in Test 2 (Clustering with Word Clouds) indicated the effectiveness of this method over the use of a Keyword Search. Thus, this method was integrated into the proposed framework.

8.4. PoC Experiment Results

The experiment results for the PoC Experiment with the developed PoC tool (named Probo) and traditional forensic tools (Encase/dtSearch) are shown in (Table 8.6). The averages for all participants during testing are shown in (Table 8.7).

Tool	Encase/dtSearch (Traditional)						Probo (Developed Tool)					
Participant	1		2		3		1		2		3	
Relevant?	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No
Error Rate	0.042	0.042	0.306	0.306	0	0	0.111	0.111	0.014	0.014	0	0
Error Rate Avg.	4%		31%		0%		11%		1%		0%	
Precision	0.974	0.939	1	0.600	1	1	0.970	0.821	1	0.970	1	1
Recall	0.950	0.969	0.436	1.000	1	1	0.821	0.970	0.975	1	1	1
F-Measure	0.962	0.954	0.607	0.750	1	1	0.889	0.889	0.987	0.985	1	1
Accuracy Avg.	0.96		0.69		1.00		0.89		0.99		1.00	
Time Spent to Complete Task (Minutes) Per Test	55		93		59		28		42		28	
Avg. Time Spent Per Tool (Minutes)	69						33					
Percentage of Change*(-1)	((33-69)/ABS(69)) * (-1) = 53%											

Table 8.6: Experiment Results for Each Participant (Encase vs. Probo)

	Probo (All Participants Avg.)			
	Relevant	Not Relevant	Total (Avg.)	Total % (Avg.)
Error Rate	0.04167	0.04167	0.04167	4%
Precision: TP/(TP+FP)	0.98990	0.93007	0.95998	96%
Recall: TP/(TP+FN)	0.93184	0.98990	0.96087	96%
F-Measure: (2*Precision*Recall)/(Precision + Recall)	0.95874	0.95783	0.95829	96%
Avg. Accuracy = 1 - Avg. Error Rate	0.95833			96%
Avg. Overall Time Spent to Complete Task (Minutes)	33		33 (Minutes)	53%
	Encase (All Participants Avg.)			
	Relevant	Not Relevant	Total (Avg.)	Total % (Avg.)
Error Rate	0.11574	0.11574	0.11574	12%

Precision: TP/(TP+FP)	0.99145	0.84646	0.91896	92%
Recall: TP/(TP+FN)	0.79530	0.98958	0.89244	89%
F-Measure: (2*Precision*Recall)/ (Precision + Recall)	0.85639	0.90128	0.87884	88%
Avg. Accuracy = 1 - Avg. Error Rate	0.88426			88%
Avg. Overall Time Spent to Complete Task (Minutes)	69		69 (Minutes)	

Table 8.7: Experiment Results for Each Tool (Combined Total % Avg.)

8.5. Interviews & Questionnaire Results

This section highlights the most interesting information and excerpts from the participants' interviews, while the detailed transcripts of the interviews are shared in (Appendix 4.2.1). As follows:

1. All the participants had a Master of Science degree in Information Security (Specializing in Cyber Security) and over 8 years of work experience in the DF field and Law Enforcement.
2. Depending on the case type and the size of the device examined, an examiner might have to review thousands of text-based files.
3. All participants stated that they rely on a Keyword Search or filtering techniques to reach their findings.
4. When asked about a files' grouping feature in traditional forensic tools most of the participants stated that they rely on the Keyword Search technique, and sometimes MD5 hash values are used to find identical files.
5. Keyword Search and the MD5 hash values matching techniques come with their own limitations. For example, any minor changes to a file results in an MD5 change, and thus files might be missed.
6. When asked about how in traditional forensic tools the information collected from a closed case could be used in a new investigation all the participants deferred to the use of

a Keyword List that could be generated from the closed case. Another technique was the MD5 hash value to find identical files. And an interesting idea was shared, which was that a Word Synonyms (i.e. words with the same meaning) technique could be used to find words related to the query.

7. When asked to share their thoughts on a method that information gathered from a closed case be used in a new investigation, most of the participants suggested that a database be built where the cases would be categorized based on the case type. For example, in a drug case a database would contain information from closed cases, then when a new drugs case is started with the database running all similarities with the old cases will be matched.
8. Another suggestion was that a tool is built that creates a list of categorized topics based on the case type, for example, information based on “names, contacts, usernames, locations, and other related information per case”. (Major Wafa Naser)
9. Final thoughts from one of the participants was that if a researcher was to create a solution they should target one “that can help us find the same text file’s content or the same words that’s been used in different text files because they would make our lives easier”. (Capt. Ghayda Ali)

The participants were asked to complete a short survey²⁰ of their opinion on the developed tool after the experiments. In addition, a short questionnaire was used to measure the level of user satisfaction; to supplement the preliminary results of the interviews.

The questionnaire results indicated that the participants agreed on the quality of the developed tool with an overall rating of 5 out of 5 on all criteria. Finally, through a survey of close-ended questions, the participants were asked to share their views based on their tests of the developed PoC tool. Their replies were as follows:

²⁰The short survey and questionnaire results are shared in Appendix 4.2.

1. All the participants shared that they thought the developed tool led them to useful results faster than other traditional forensic tools they had used.
2. When asked if they had encountered similar grouping techniques in any other tool, the shared view was that they had only seen such a technique in the developed tool. Additionally, it was noted by one of the participants that even though most traditional forensic tools offer the Keyword Search feature “the tested tool was much better and faster and easier to use” (Capt. Ghayda Ali).
3. All participants were in agreement that the developed tool and the techniques it is promoting could be a useful addition to have in current forensic tools.
4. Finally, when asked for suggestions to improve on the developed tools’ techniques, one of the participants suggested that similarity techniques for pictures, video, and audio files be created so they could build a database of relevant files for cases of similar types.

Chapter Nine: Discussion

This chapter shares the analysis and interpretation of the results. The first section of this chapter talks about the experiments results. The second shares the interviews interpretations. The third discusses the key findings as they relate to the research aims. The last section outlines some of the problems and observations that were encountered during the experiments, and attempts to provide technical reasons for each.

9.1. Discussion of Experiments

The results of the Clustering Component Experiment were straight forward. The K-Means algorithm gave an average accuracy of 97.1%, while the other algorithms gave 95.2% with Hierarchical Clustering and 88.9% with Random Clustering. Based on these results, K-Means was selected for the clustering component of the proposed framework.

This proved the results of the study by Pallavi, et al. (2014), as K-Means achieved better results when it was properly initialized. This initialization could also be achieved while using the PoC, if the user was able to correctly estimate the number of clusters within the dataset.

The results of the Classification Component Experiment indicated that the kNN algorithm had performed better than the other two during the Training Stage. The average accuracy for kNN reached 97.5%, while NB scored 97.25% and SVM scored 96.64%. On the other hand, kNN matched the NB algorithm's performance during the Testing Stage with an average accuracy of 97.2%, whereas SVM dropped to 96.3%. Therefore, based on these results, kNN algorithm was selected for the proposed framework.

In the Search & Visualization Experiment, comparing the Keyword Search technique and clustering with Word Clouds showed that the use of the latter appeared to provide an advantage in terms of speed. The Percentage of Change²¹ measure calculates the percentage of change between values according to Microsoft (2012).

Based on the results, searching with text clustering and Word Clouds was 76% faster than Keyword Search. The combination of text clusters and visualization simplifies the search process, given a specific target for the search.

An interesting observation was that the number of clusters displayed to the user played an important role in their navigation between the clusters. For example, given 10 clusters it might have taken longer, than if they were given 6 clusters, to reach the correct file to answer the questions during the experiment. In other words, having a narrower scope to look at made for a faster route to the answers.

Another idea that was of interest was the mapping (using visualized links) between the generated clusters. However, the end result of this particular idea was a large mesh of interconnected clusters. And while the idea had merits, it was difficult to apply in practice as the main purpose of this part of the study was to find a better method to sift through document collections of unknown contents.

In real world DF investigations, the use of the proposed text clustering with word cloud visualization methods might lack the dynamics of keyword searches. In contrast to the current

²¹ Further details can be found in Appendix 4.1.5: Percentage of Change.

reliance of real investigations on keyword search techniques, the idea of an integrated solution that combines text clustering, keyword searches and word clouds could be more appealing to an investigator. Hence, this research combined the proposed visualization method with Text Data Mining (TDM) techniques and used real world criminal cases for its testing purposes.

The PoC Experiment gave encouraging results. The PoC provided on average lower error rates with an improvement of 8%, better average precision with an improvement of 4%, better average recall with an improvement of 7%, and finally better overall average accuracy of 96% compared to the traditional tool's average accuracy of 88%, as shown in (Figure 9.1).

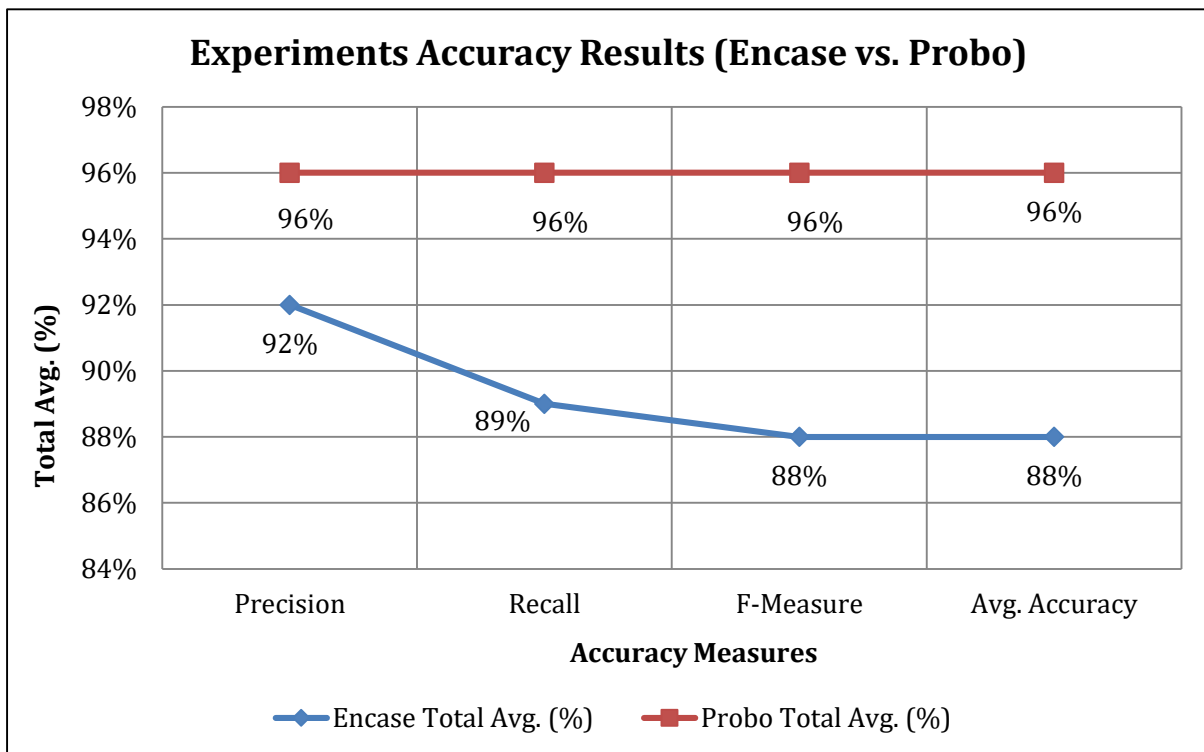


Figure 9.1: Experiments Accuracy Results (Encase vs. Probo)

Based on the results, the proposed techniques in Probo gave better average accuracy results, with a single exception being Participant 1 where they scored higher on Encase (96%) than Probo (89%). On the other hand, Probo gave the results faster, as shown in (Table 9.1).

	Encase Total % (Avg.)	Probo Total % (Avg.)
Error Rate	12%	4%
Precision	92%	96%
Recall	89%	96%
F-Measure	88%	96%
Accuracy Avg.	88%	96%
Avg. Overall Time for Completion	69 Minutes	33 Minutes

Table 9.1: Experiment Result Total % Averages (Participants Combined Avg.)

The overall time spent on average to complete the tasks indicates that the proposed techniques, as shown by Probo, decreased the average required time by 53% over the time spent using Encase²². Specifically, the average time spent to answer the Keyword Search related questions decreased by 88% over the average time spent using Encase. Moreover, the average time spent to answer the Relevancy question decreased by 18% over the time spent using Encase.

9.2. Discussion of Interviews

The participants of this study work in the Emirates Electronic Evidence Center (EEEC) at Abu Dhabi Police, in the UAE. The Center was commissioned towards the end of 2009 to facilitate DF

²² Further details can be found in Appendix 4.1.5: Percentage of Change.

investigations to different entities internally within the Police and externally for the Ministry of Justice (MoJ) by serving different UAE Courts.

The DF senior investigators of EEEEC are among the first individuals in the UAE to be accredited under the UAE's Ministry of Justice (MoJ) as Expert Witnesses in the field of Digital Evidence.

Over the years, the EEEEC staff handled hundreds of DF cases for traditional (i.e. homicides and narcotics) and cyber (i.e. hacking, intrusion, and fraud) crimes.

The selected participants of the study all had experience dealing with DF text-based heavy data analysis that pertained to various case types. The purpose of interviewing the participants was to ensure the validity and reliability of the study by taking other DF investigators views into account.

The participants indicated that most of the cases they had dealt with previously required the review of thousands of text files. They also indicated that the grouping of similar findings as a feature is lacking when it comes to the DF tools they relied on. Furthermore, they stated that keyword searching and MD5 hash matching techniques were two of their primary search methods to look for similar files.

This implied that the analysis of thousands of files is left to the DF investigator to manually process with the assistance of these search techniques. This further supported the identified research gap²³ when it comes to the challenges in the DF field.

²³ Further details can be found in Section 2.1.4.4: Research Gap.

Furthermore, one of the participants suggested that a collection of files of interest from one case could be used for similar cases in the future. This is an important issue, which was implemented in the proposed solution.

An interesting idea for future work was the use of filenames and other metadata as an added search vector. This idea had been used in the studies by Fahdi, et al. (2016) and Mohammed, et al. (2016). However, for future implications in relation to this study, the use of metadata of files of interests would theoretically speed up the analysis process and increase accuracy.

9.3. Key Findings

Accurate performance evaluation of any system is the key to its improvement. Even though, the developed PoC achieved on average better accuracy, with a single exception being Participant 1, the PoC implementation showed encouraging results.

The PoC on average reached 96% of Precision, Recall, F-Measure, and Accuracy. Furthermore, it demonstrated that it can be used to reduce the human analytical time by 53% on the tested dataset. In addition, there were several important findings, including:

- 1.** In terms of identifying the best algorithms to integrate into the proposed solution, K-Mean and kNN gave the best results as clustering and classification algorithms, respectively.
- 2.** The use of Word Clouds to visualize clusters was a better search method than Keyword Search. This technique can play an important role in visualizing clustered information.
- 3.** The developed framework was based on the integration of the tested techniques from the aforementioned steps. The proposed solution, as seen during the PoC Experiment, improved both average accuracy and speed when compared to traditional DF tools.

4. The degree of improvement was tested using the evaluation measurements. The end results indicated that the use of an examiner's relevancy feedback can help speed up analysis when it comes to a text-based heavy investigation.

9.4. Observations

The users pointed out some issues they had encountered during the PoC Experiment, while the researcher observed some other issues. The following list highlights those observations and provides technical reasons for each.

1. Issue 1: Dataset Arabic Text (Encase)

a. Observations:

- i. During the experiment involving the use of Encase all participants encountered a problem when trying to use the Encase's Keyword Search feature. The Arabic text was displayed in Encase Left-To-Right (LTR), while the given files themselves were all Right-to-Left (RTL).
- ii. The participants attempted to use different search expressions and syntax, but none helped in Encase (i.e. using Unicode/UTF-8 did not help either).
- iii. One of the participants attempted to search for certain keywords by flipping the Arabic text (LTR), with no results found.
- iv. As a solution to this problem two of the participants decided to use another tool (dtSearch v.7.07), while the other relied on Encase and the manual preview of files.

b. Reasons:

- i. The dataset for all tests were created from text files (Microsoft Word 2007) that were converted into PDF files. The problem could be attributed to how Arabic text is saved to PDF.

2. Issue 2: Misclassification of Files (Participant 1)

- a. **Observation:** during the experiment involving the use of Probo, Participant 1 indicated that about 15 files were classified as Relevant, but were not upon further analysis by the user.
- b. **Reason:** it was observed that the participant had chosen to create 4 clusters only, and classifying 3 out of 4 as “Relevant” and 1 as “Not Relevant”. The issue of this misclassification can be attributed to the number of files that were grouped under “Relevant”, but in reality were not. Thus, because of the low number of clusters (i.e. 4) in a collection of 72 files and over 5 different categories in the dataset (unbeknown to the user), the tool had misclassified those indicated files. Furthermore, the user was able to view each of the files to choose their relevancy afterwards.

3. Issue 3: Misclassification of Files (Participant 2)

- a. **Observations:**
 - i. During the experiment involving the use of Probo, Participant 2 indicated that 1 file had been misclassified as “Relevant”, but in fact the file was not.
 - ii. It was observed that Participant 2 had chosen to create 7 clusters, which contributed to the low number of False Positive (only 1 file misclassified by the tool).
- b. **Reason:** this misclassification can be attributed to the file content falling within range of similar files in a cluster that was labeled as “Relevant”. Furthermore, the user was able to view each of the files to choose their relevancy afterwards.

4. Issue 4: Selection Error (Participant 2)

- a. **Observation:**
 - i. During the experiment involving the use of Probo, it was observed that Participant 2 viewed a file that was “Relevant” (also labeled by Probo as “Relevant”) and the contents could be said to be relevant to the request. However, the user did not select it as being “Relevant”.

- ii. It is also worth mentioning that the experiment accuracy results for Probo for this Participant fell from 100% to 99%, because of this missed file (False Negative).
- b. **Reason:** this can be attributed to human error.

5. **Issue 5: Developed Tool Bugs**

- a. **Observation:** during the experiment involving the use of Probo, it was observed that the tool had a few bugs that needed to be fixed on the side of the User Interface (UI), where it showed a duplication of the files in the Classification Results page.
- b. **Reason:** this can be attributed to bugs that were not encountered during the development and testing of the UI. Moreover, since it did not disrupt the experiments for the participants, as indicated by them, the testing was not stopped and the users were able to use the tool without problems.

Chapter Ten: Conclusion & Recommendations

This chapter discusses the study conclusion, possible limitations, recommendations, and future work.

10.1. Conclusion

The general focus of this study was on the Digital Forensics (DF) discipline, and it specifically targeted the issues of the high information retrieval overhead and the increase of DF examination times. Thus, it presented a framework that integrates Text Data Mining (TDM) techniques and an investigator's Relevancy Feedback (RF) to help reach useful information faster.

The framework was evaluated using a text-based dataset comprised of Arabic text taken from real world criminal cases. Even though this work focused²⁴ on Arabic text, the proposed framework is language independent.

The study showcased the potential of the proposed framework through a Proof-of-Concept (PoC) tool that was tested during experiments with senior DF investigators from the Emirate Electronic Evidence Center (EEEC). In addition, the goal of these experiments was to compare the results of using the developed solution against traditional DF tools by expert DF investigators.

The experimental results imply the usefulness of the proposed framework and potential of increasing the accuracy and speed of analysis results during a DF investigation. Furthermore, the proposed solution overcame some of the problems associated with traditional search techniques,

²⁴ The reason for this focus was explained in Section 6.2

i.e. MD5 and keyword search methods. These problems were highlighted during the conducted interviews with the participants and in the literature by Breitinger & Roussev (2014).

The proposed framework was an attempt to answer the researcher question, while the PoC was an example of how it could be implemented. In order to answer the research question it was broken down into three main aims, which led to the key findings of this study and eventually the creation and testing of the framework.

The first aim established that there was a lack of a file's grouping technique in current DF tools and that relevant information found during the analysis can be used in a limited way and only manually by the investigator. These findings indicated the need of such techniques to help the investigator reach results faster, independent of the listed limitations during the participants' interviews. Moreover, it indicated that relevancy feedback from an investigator is still an untapped resource when it comes to established DF tools.

The second aim resulted in the creation of the framework that proposed the incorporation of TDM and RF into the digital forensics investigation. The framework's main components were tested separately using different experiments. The results indicated that K-Means performed better than the other tested clustering algorithms, while kNN gave the best results when compared to the other classification algorithms. Additionally, the technique of using Word Clouds to visualize findings was a better search method than Keyword Search.

The third aim confirmed the significance of the proposed framework through the implemented PoC, by testing it on a real world dataset. The implementation obtained reasonably good results; by increasing the average accuracy and speed of reaching useful information. The involvement of

experienced senior DF investigators in the evaluation of the PoC implementation was an added value to the potential of the proposed solution.

Finally, based on the findings and achieved results of this study it is safe to say that the proposed framework represents a potential answer to the main research question. It provided a solution that combined TDM and RF techniques. Furthermore, the experiments showed that it decreased the false positive (and false negative) rates of an examination, improved the recall-precision results, and shortened the time needed to complete the tasks by reducing the burden on the DF investigator.

10.2. Limitations

Based on the discussed findings and results it is safe to say that this research provides a promising answer to the posed problem. While this study could be seen as a good foundation for future research, it comes with its own limitations.

The first limitation is the high computational expense that should be expected based on the dataset size. And while the experiments datasets were processed fairly quickly because of their small size, real cases contain a significant amount of data that should be kept in mind.

The rationale behind keeping the test set small can be attributed to two main reasons. The first reason was that while a large number of files from different real world cases were available, there was a limited allotted time with the expert participants, which was distributed between interviews and practical tool testing.

Therefore, having them sift through hundreds of files was not feasible. The second reason was that the developed PoC tool relied on a number of tools and had a limitation on the number of

items that could be uploaded without hitting the data complexity limit for the Educational License for those tools.

The second limitation is that the number of participants was small. The study targeted DF senior examiners of EEEEC who were among the first individuals in the UAE to be accredited under the UAE's Ministry of Justice (MoJ) as Expert Witnesses in the field of Digital Evidence.

At the time of this study there were only three participants who fit the profile and were available to participate. Additionally, the dataset used for the participant's experiments all contained confidential information that only Senior DF examiners within the EEEEC were allowed to preview as part of their normal jobs. It is worth noting that the combined experience of these participants was over 24 years in the DF field.

The third limitation is that while the proposed solution achieved faster results, when compared to the traditional DF tools, it still does not replace those tools. For the simple reason that forensic tools target several aspects within the investigative process, i.e. file carving, file signature analysis, or entropy analysis. Whereas, the proposed solution focused on the text-based document analysis only, so it shall not be thought of as a standalone solution to the problem. Conversely, it should be considered as an efficient add-on to investigations where text-based documents are deeply scrutinized.

The fourth limitation is that even though the tool achieved better results when the number of created clusters was higher (i.e. during Participants 2 and 3 testing of the developed tool), the

number of misclassified files²⁵ indicates that further research might be required to better optimize the number of clusters. There were two types of misclassifications, human based (by the users) and tool based (by the developed tool).

The first type can be linked to human judgment or error, while the second type is based on the similarity between objects in the labeled clusters. Additionally, the results showed that the more clusters were chosen to be created, the more accurate the results were. In other words, the less objects were in each cluster, the more accurate the clustering.

The last limitation and most obvious is the generalization of the end results, for the reason that no two criminal cases are identical in a DF investigation setting. Cases are, to a certain extent, more similar than not based on the criminal case type.

Therefore, intelligence gathered from closed cases can be used between different cases of similar case types. Additionally, as indicated by the participant's answers during the interviews to the question of how information collected from a closed case can be used in a new investigation.

Their answers²⁶ were almost identical, by using the following techniques:

- “...hash values, filenames, and keyword list” (Capt. Abdalla Al Ali)
- “...Keyword Search. For example, if I have a drug case and I use certain words in the case to find some files or some emails, I can use the same Keyword Search for future cases. But if I create MD5 hashes for some of those files I will not be able to find something that had been altered with as little change as a word within the file. Because the MD5 is looking at

²⁵ Further details can be found in Section 9.4: Observations (Issue 2).

²⁶ Further details can be found in Appendix 4.2.1: Interview Combined Transcripts.

the whole file that has the same MD5, so if there was any change on it the MD5 for that file will be different.” (Capt. Ghayda Ali)

- “...Keyword list and hash values, however it's not always accurate since that hash value could be changed if the file was altered.” (Major Wafa Naser)

Finally, the developed solution overcame some of those forensic tools' shortcomings by making similarity of the files' contents its focus. However, more tests might be required to examine the effectiveness of the solution between a broader set of similar and dissimilar case types.

10.3. Recommendations

Many researchers and software developers focus on helping the DF investigator find results faster through data preprocessing automation methodologies prior to the beginning of the analysis phase, while this study focused more on "during" the analysis phase. This work could be adopted by forward thinking DF tool's developing companies as part of their future system designs.

This research discussed different data handling techniques and proved their usefulness in the context of a DF investigation. For example, the use of TDM techniques to carry forward intelligence gathered from closed cases and their relevancy to the investigation, which could be used in future similar investigations. This could be achieved through the use of the Relevant Database component.

Moreover, the other system component added was the Clustering of documents based on their contents and possible relationships, which can be a valuable instrument to a DF investigator of digital evidence. For the examiner to be able to reach targeted information, they could rely on the

Word Cloud component, which provides a simple and powerful tool that constitute faster data examinations and can be supplemented with a deeper file examination.

10.4. Future Work

Future work will focus on developing a centralized shared database for the different case categories, which could be updated and used by different users at the same time. Additionally, further work could be undertaken to investigate the possibility of adding an Agent component to the framework. The role of this Agent would be to facilitate real-time sharing of intelligence between the different examiners working on the same case.

Moreover, future work will look into adding tags for case categories by using a predefined term list that identifies relevant files that fall under specific categories. For example, different case profiles could be created and automatically used by the investigator to identify and tag possibly related files.

Furthermore, the presented framework has a good foundation as an intelligence sharing platform. While it might be beyond the scope of digital forensics work, finding a way of sharing confidential information of similar cases between law enforcement agencies across different jurisdictions would be of great value. Such work would involve finding a way to encrypt the content of the database, whilst it remains accessible to the intelligence platform.

The development of a shared criminal cases' database from DF investigations that could be used by different agencies without giving away access into the confidential aspects of cases has merit. Finally, this might help promote the sharing of timely intelligence confidentially, as only the tool

would be able to interpret the shared data and find similar traces in certain cases under investigation.

References

- Abdelaal, H. M., Elmahdy, A. N., Halawa, A. A. & Youness, H. A., 2018. Improve the Automatic Classification Accuracy for Arabic Tweets Using Ensemble Methods. *Journal of Electrical Systems and Information Technology*, 5(3), pp. 363-370.
- Abooraig, R. et al., 2018. Automatic Categorization Of Arabic Articles Based On Their Political Orientation. *Digital Investigation*, Volume 25, pp. 24-41.
- ACPO, A. o. C. P. O., 2005. *Good Practice Guide for Computer-Based Electronic Evidence*. [Online] Available at: https://www.7safe.com/docs/default-source/default-document-library/acpo_guidelines_computer_evidence_v4_web.pdf [Accessed 01 08 2018].
- Adelstein, F., 2006. Live Forensics: Diagnosing your System Without Killing it First. *Commun ACM*, 49(2), pp. 63-66.
- Alakrot, A., Murray, L. & Nikolov, N. S., 2018. Towards Accurate Detection of Offensive Language in Online Communication in Arabic. *Procedia Computer Science*, Volume 142, pp. 315-320.
- Al-Anazi, S., AlMahmoud, H. & Al-Turaiki, I., 2016. *Finding Similar Documents Using Different Clustering Techniques*. In: *Symposium on Data Mining Applications, SDMA2016*. Riyadh, Elsevier.
- Al-Anzi, F. S. & AbuZeina, D., 2016. *Big Data Categorization for Arabic Text Using Latent Semantic Indexing and Clustering*. Bangkok, International Conference on Engineering Technologies and Big Data Analytics (ETBDA'2016).
- Alawadhi, I., Read, J. C., Marrington, A. & Franqueira, V. N. L., 2015. Factors Influencing Digital Forensic Investigations: Empirical Evaluation Of 12 Years Of Dubai Police Cases. *Journal of Digital Forensics, Security and Law*, 10(4), pp. 6-16.
- Albalate, A., Suchindranath, A. & Minker, W., 2010. *A semi-supervised cluster-and-label approach for utterance classification*. Kuala Lumpur, INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association.
- Alkoffash, M. S., 2012. Comparing between Arabic Text Clustering using K Means and K Mediods. *International Journal of Computer Applications (0975 – 8887)*, 51(2).
- Al-Omari, O. M., 2011. Evaluating the Effect of Stemming in Clustering of Arabic Documents. *Academic Research International*, 1(1), p. 284–291.

- Alruily, M., Ayes, A. & Al-Marghilani, A., 2010. *Using Self Organizing Map to Cluster Arabic Crime Documents*. Wisla, Poland, Proceedings of the International Multiconference on Computer Science and Information Technology, p. 357–363.
- Alsaleem, S., 2011. Automated Arabic Text Categorization Using SVM and NB. *International Arab Journal of e-Technology*, 2(2), pp. 124-128.
- Al-Sarrayih, H. S. & Al-Shalabi, R., 2009. *Clustering Arabic Documents Using Frequent Item Set-Based Hierarchical Clustering with an N-Grams*. Amman, Jordan, The International Conference on Information Technology.
- Al-Shalabi, R., Kanaan, G. & Gharaibeh, M. H., 2006. *Arabic Text Categorization Using kNN Algorithm*. Amman, Jordan, Proceeding of the 4th International Multiconference on Computer Science and Information Technology.
- Alwidian, S. A., Bani-Salameh, H. A. & Alsaity, A. N., 2015. Text data mining: a proposed framework and future perspectives. *International Journal of Business Information Systems*, 18(2).
- Alzaabi, M., Jones, A. & Martin, T. A., 2013. An Ontology-Based Forensic Analysis Tool. *Annual ADFSL Conference on Digital Forensics, Security and Law*, Volume 5, p. 121–135.
- Ayers, D., 2009. A Second Generation Computer Forensic Analysis System. *Digital Investigation*, Volume 6, pp. S34-S42.
- Baeza-Yates, R. & Ribeiro-Neto, B., 2011. *Modern Information Retrieval*. 2nd ed. Edinburgh Gate, Essex: Pearson Education Limited.
- Banin, S. & Dyrkolbotn, G. O., 2018. Multinomial Malware Classification via Low-Level Features. *Digital Investigation*, Volume 26(Supplement), pp. S107-S117.
- Bayne, E., Ferguson, R. & Sampson, A., 2018. *OpenForensics: A Digital Forensics GPU Pattern Matching Approach for the 21st Century*. In: *Proceedings of the Fifth Annual DFRWS Europe*. Florence, Elsevier Ltd.
- Bechini, A., Marcelloni, F. & Segatori, A., 2016. A MapReduce Solution for Associative Classification of Big Data. *Information Sciences*, Volume 332, pp. 33-55.
- Beebe, N. & Clark, J., 2006. *Dealing with Terabyte Data Sets in Digital Investigations*. In: Pollitt M., Sheno S. (eds) *Advances in Digital Forensics*. Boston, MA, Springer.
- Beebe, N. & Dietrich, G., 2007. A New Process Model for Text String Searching. In: C. P. & S. S., eds. *Advances in Digital Forensics III*. Orlando: Springer, pp. 179-191.
- Beebe, N. L., 2009. Digital Forensic Research: The Good, the Bad and the Unaddressed. In: G. Peterson, ed. *Advances in Digital Forensics V*. Florida: Springer-Verlag Berlin and Heidelberg GmbH & Co. K, pp. 17-36.

- Beebe, N. L., 2013. *SIFTER: Search Indexes For Text Evidence Relevantly*. Chantilly, VA, Open Source Digital Forensics Conference (OSDFCON). [Accessed 27 March 2016]. Available at: http://osdfcon.basistech.wpengine.com/presentations/2013/Nicole_Beebe.pdf.
- Beebe, N. L. & Clark, J. G., 2007. Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results. *Digital Investigation: The International Journal of Digital Forensics & Incident Response*, 4(1), p. 49–54.
- Beebe, N. L. et al., 2011. Post-retrieval search hit clustering to improve information retrieval effectiveness: Two digital forensics case studies. *Decision Support Systems*, 51(4), pp. 727-920.
- Bertè, R., Marturana, F., Me, G. & Tacconi, S., 2012. *Data Mining based Crime-Dependent Triage in Digital Forensics Analysis*. Taipei, Taiwan, International Conference on Affective Computing and Intelligent Interaction (ICACII 2012).
- Bharadwaj, N. K. & Singh, U., 2018. Efficiently Searching Target Data Traces In Storage Devices With Region Based Random Sector Sampling Approach. *Digital Investigation*, Volume 24, pp. 128-141.
- Bhat, V. H. et al., 2010. A Data Mining Approach for Data Generation and Analysis for Digital Forensic Application. *IACSIT International Journal of Engineering and Technology*, 2(3), pp. ISSN: 1793-8236.
- Blokh, I. & Alexandrov, V., 2017. News Clustering Based on Similarity Analysis. *Procedia Computer Science*, Volume 122, pp. 715-719.
- Bolle, T. & Casey, E., 2018. Using Computed Similarity of Distinctive Digital Traces to Evaluate Non-Obvious Links and Repetitions in Cyber-Investigations. *Digital Investigation*, 24(Supplement), pp. S2-S9.
- Breiman, L., 1996. Bagging Predictors. *Machine Learning*, Volume 24, pp. 123-140.
- Breitinger, F. & Roussev, V., 2014. Automated Evaluation of Approximate Matching Algorithms on Real Data. *Digital Investigation*, Volume 11, pp. S10-S17.
- Bsoul, Q., Salim, J. & Zakaria, L. Q., 2013. *An Intelligent Document Clustering Approach to Detect Crime Patterns*. Malaysia, Elsevier Ltd., pp. 1181-1187.
- Buckley, C., Salton, G. & Allan, J., 1994. *The effect of adding relevance information in a relevance feedback environment*. Dublin, Ireland, Springer-Verlag, pp. 292-300.
- Burnap, P., French, R., Turner, F. & Jones, K., 2018. Malware Classification using Self Organising Feature Maps and Machine Activity Data. *Computers & Security*, Volume 73, pp. 399-410.
- Carrier, B. & Grand, J., 2004. A Hardware-Based Memory Acquisition Procedure For Digital Investigations. *Digital Investigation*, 1(1), pp. 50-60.
- Casey, E., 2010. *Handbook of Digital Forensics and Investigation*. 1st ed. San Diego: Elsevier Inc.

- Casey, E., 2011. *Digital Evidence and Computer Crime: Forensic Science, Computers and the Internet*. 3rd ed. California: Elsevier Ltd.
- Cherkassky, V., 1995. *The Nature Of Statistical Learning Theory*. New York, IEEE Transactions on Neural Networks.
- Cohen, M., Garfinkel, S. & Schatz, B., 2009. Extending the Advanced Forensic Format to Accommodate Multiple Data Sources, Logical Evidence, Arbitrary Information and Forensic Workflow. *Digital Investigation*, 6(Supplement), pp. S57-S68.
- Council, N. R., 2009. *Strengthening Forensic Science in the United States: A Path Forward*. 1 ed. Washington, DC: The National Academies Press.
- Creamer, G. G., Stolfo, S. & Hershkop, S., 2006. *A Temporal Based Forensic Analysis of Electronic Communication*. San Diego, CA, Digital Government Proceedings.
- Creswell, J. W. & Clark, V. L. P., 2007. *Designing and Conducting Mixed Methods Research*. 1st ed. London: Sage Publications Ltd.
- Dahiya, P. & Srivastava, D. K., 2018. Network Intrusion Detection in Big Dataset Using Spark. *Procedia Computer Science*, Volume 132, p. 253–262.
- Dataflair, 2018. *What is Text Mining in Data Mining – Process & Applications*. [Online] Available at: <https://data-flair.training/blogs/text-mining/> [Accessed 25 01 2019].
- Decherchi, S. et al., 2009. *Text Clustering for Digital Forensics Analysis*. Burgos, Spain, Computational Intelligence in Security for Information Systems - CISIS'09, 2nd International Workshop Proceedings.
- Department of Justice, U., 2016. *Audit of the Federal Bureau of Investigation's Philadelphia Regional Computer Forensic Laboratory*, Pennsylvania: Office of the Inspector General U.S. Department of Justice.
- Eisenhardt, K. & Graebner, M., 2007. Theory Building from Cases: Opportunities and Challenges. *Academy of Management Journal*, 50(1), pp. 25-32.
- El Kourdi, M., Bensaïd, A. & Rachidi, T., 2004. *Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm*. Geneva, Proceedings of the Workshop on Computational Approaches to Arabic.
- Elder, J. et al., 2012. *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*. 1st ed. Oxford, UK: Academic Press.
- Facebook-IQ, 2016. *The Multidevice Movement: Teens in France and Germany*. [Online] Available at: <https://www.facebook.com/business/news/insights/the-multidevice-movement-teens-in-france->

and-germany

[Accessed 10 05 2019].

Fahdi, M. A., Clarke, N., Li, F. & Furnell, S., 2016. A Suspect-Oriented Intelligent and Automated Computer Forensic Analysis. *Digital Investigation*, Volume 18, pp. 65-76.

Farghaly, A. & Shaalan, K., 2009. Arabic Natural Language Processing: Challenges and Solutions. *ACM Transactions on Asian Language Information Processing (TALIP)*, 8(4), p. 14.

Fei, B., Eloff, J. H. P., Venter, H. & Olivier, M. S., 2005. *Exploring Forensic Data with Self-Organizing Maps*. Orlando, Florida, Advances in Digital Forensics, IFIP International Conference on Digital Forensics.

Forte, D., 2004. The Importance of Text Searches in Digital Forensics. *Network Security*, 4(1), p. 13–15.

Frederick, U. & Christiana, O., 2017. An Improved Data Mining Tools Classifier for Digital Forensic Analysis. *Electroscope Journal*, 8(8), pp. 117-128.

Freiling, F. & Zoubek, C., 2017. Do Digital Investigators Have to Program? A Controlled Experiment in Digital Investigation. *Digital Investigation*, 20(Supplement), pp. S37-S46.

García-Pablo, A., Cuadros, M. & Rigau, G., 2018. W2VLDA: Almost Unsupervised System for Aspect Based Sentiment Analysis. *Expert Systems With Applications*, Volume 91, p. 127–137.

Garfinkel, S., 2012. Lessons Learned Writing Digital Forensics Tools and Managing a 30 TB Digital Evidence Corpus. *Digital Investigation*, Volume 9, p. S80–S89.

Garfinkel, S. L., 2007. *Carving contiguous and fragmented files with fast object validation*. Pittsburgh, PA, Elsevier Ltd.

Garfinkel, S. L., 2010. *Digital forensics research: The next 10 years*. Monterey, Elsevier Ltd, p. S64–S73.

Gharib, T. F., Habib, M. B. & Fayed, Z. T., 2009. Arabic Text Classification Using Support Vector Machines. *International Journal of Computer Applications - IJCA*, 16(4), pp. 192-199.

Gholap, P. & Maral, V., 2015. Information Retrieval of K-Means Clustering for Forensic Analysis. *International Journal of Science and Research (IJSR)*, 4(1), pp. 577-581.

Ghwanmeh, S. H., 2007. Applying Clustering of Hierarchical K-means-like Algorithm on Arabic Language. *International Journal of Computer and Information Engineering*, 1(8), pp. 2396-2400.

Glesne, C., 2011. *Becoming Qualitative Researchers: An Introduction*. 4th ed. Boston: Pearson.

Gogolin, G., 2010. The Digital Crime Tsunami. *Digital Investigation*, 7(1–2), pp. 3-8.

- Gormley, C. & Tong, Z., 2015. *Pluggable Similarity Algorithms*. [Online]
Available at: <https://www.elastic.co/guide/en/elasticsearch/guide/current/pluggable-similarities.html>
[Accessed 25 08 2018].
- Grajeda, C., Breiting, F. & Baggili, I., 2017. Availability of Datasets for Digital Forensics – and What is Missing. *Digital Investigation*, Volume 22, pp. S94-S105.
- Grobler, C. P., 2011. *A Digital Forensic Management Framework [Thesis]*. [Online]
Available at: <https://ujdigispace.uj.ac.za/bitstream/handle/10210/6365/Grobler.pdf?sequence=1>
- Guan, L., He, Y. & Kung, S.-Y., 2017. *Multimedia Image and Video Processing*. 2nd ed. Boca Raton, FL: CRC Press.
- Hadjidj, R. et al., 2009. Towards an integrated e-mail forensic analysis framework. *Digital Investigation: The International Journal of Digital Forensics & Incident Response*, 5(3-4), pp. 124-137.
- Hadni, M., Ouatik, S. A. & Lachkar, A., 2013. Effective Arabic Stemmer Based Hybrid Approach for Arabic Text Categorization. *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, 3(4), pp. 1-14.
- Hakim, C., 2000. *Research Design: Successful Designs for Social Economics Research*. 2nd ed. New York: Routledge.
- Halboob, W. et al., 2015. *Data Warehousing Based Computer Forensics Investigation Framework*. Las Vegas, US, 12th International Conference on Information Technology - New Generations, pp. 163-168.
- Harman, D., 1992. Relevance feedback and other query modification techniques. In: W. B. Frakes & R. Baeza-Yates, eds. *Information Retrieval: Data Structures and Algorithms*. NJ, USA: Prentice-Hall, Inc, pp. 241-263.
- Harman, D., 1992. *Relevance feedback revisited*. Copenhagen, Denmark, ACM, pp. 1-10.
- Hastie, T., Tibshirani, R. & Friedman, J., 2016. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Second ed. Stanford, CA: Springer Series in Statistics.
- Hearst, M. A., 1999. *Untangling Text Data Mining*. Stroudsburg, PA, Association for Computational Linguistics, pp. 3-10.
- Hearst, M. A. & Pedersen, J. O., 1996. *Reexamining the cluster hypothesis: scatter/gather on retrieval results*. New York, ACM, pp. 76-84.
- He, G., Neumayer, R. & Norvag, K., 2004. *Learning to cluster web search results*. New York, Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 210-217.
- Hlaváč, V., 2018. *Classifier performance evaluation*. [Online]
Available at: <http://people.ciirc.cvut.cz/~hlavac/TeachPresEn/31PattRecog/13ClassifierPerformance.pdf>
[Accessed 08 08 2018].

- Hmeidi, I., Hawashin, B. & El-Qawasmeh, E., 2008. Performance of KNN and SVM Classifiers on Full Word Arabic Articles. *Advanced Engineering Informatics*, 22(1), pp. 106-111.
- Honale, P. S. & Borkar, J., 2015. Framework for Live Digital Forensics Using Data Mining. *International Journal of Computer Trends and Technology (IJCTT)*, 22(3), pp. 117-121.
- Hudson, B. A., 1994. *Punishing the Poor: A Critique of the Dominance of Legal Reasoning In Penal Policy and Practice*. Manchester, Manchester University Press.
- Humphreys, K., Demetriou, G. & Gaizauskas, R., 2000. Bioinformatics applications of information extraction from scientific journal articles. *Journal of Information Science*, 26(2), pp. 75-85.
- Idrees, F. et al., 2017. Pindroid: A Novel Android Malware Detection System Using Ensemble Learning Methods. *Computers & Security*, Volume 68, pp. 36-46.
- Ingersoll, G. S., Morton, T. S. & Farris, A. L., 2013. *Taming Text*. 1st ed. NY: Manning Publications.
- Interpol, 2018. *International Child Sexual Exploitation database*. [Online]
Available at: <https://www.interpol.int/en/How-we-work/Databases/International-Child-Sexual-Exploitation-database>
[Accessed 5 5 2019].
- Inuwa-Dutse, I., Liptrott, M. & Korkontzelos, I., 2018. Detection of Spam-Posting Accounts on Twitter. *Neurocomputing*, Volume 315, p. 496–511.
- Iqbal, F., Binsalleeh, H., Fung, B. C. & Debbabi, M., 2010. Mining Writeprints from Anonymous E-Mails for Forensic Investigation. *Digital Investigation*, Volume 7, pp. 56-64.
- Iqbal, F., Hadjidj, R., Fung, B. C. M. & Debbabi, M., 2008. A Novel Approach of Mining Write-Prints for Authorship Attribution in E-mail Forensics. *Digital Investigation*, 5(Supplement), p. S42–S51.
- Ishihara, S., 2017. Strength of Linguistic Text Evidence: A Fused Forensic Text Comparison System. *Forensic Science International*, Volume 278, pp. 184-197.
- Jaybhaye, S. P., 2015. Document Clustering for Forensic Analysis: An Approach for Improving Computer Inspection. *Asian Journal of Engineering and Technology Innovation*, 8(1), pp. 46-54.
- Johansen, G., 2017. *Digital Forensics and Incident Response*. 1st ed. Birmingham, UK: Packt Publishing Ltd.
- Karbab, E. B., Debbabi, M., Derhab, A. & Mouheb, D., 2018. MalDozer: Automatic Framework for Android Malware Detection using Deep Learning. *Digital Investigation*, Volume 24(Supplement), pp. S48-S59.
- Karima, A., Loqmana, C. & Boumhidi, J., 2018. Determining the Number of Clusters using Neural Network and Max Stable Set Problem. In: The First International Conference On Intelligent Computing in Data Sciences. *Procedia Computer Science*, Volume 127, p. 16–25.

- Karystianis, G., Thayer, K., Wolfe, M. & Tsafnat, G., 2017. Evaluation of a Rule-Based Method for Epidemiological Document Classification Towards the Automation of Systematic Reviews. *Journal of Biomedical Informatics*, Volume 70, p. 27–34.
- Kaushik, S., 2016. *An Introduction to Clustering and Different Methods of Clustering*. [Online] Available at: <https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/> [Accessed 24 05 2017].
- Khammassi, C. & Krichen, S., 2017. A GA-LR Wrapper Approach for Feature Selection in Network Intrusion Detection. *Computers & Security*, Volume 70, pp. 255-277.
- Khan, M., Chatwin, C. & Young, R., 2007. A Framework for Post-Event Timeline Reconstruction Using Neural Networks. *Digital Investigation*, 4(3-4), pp. 146-157.
- Khenat, S., Kolhatkar, P., Parit, S. & Joshi, S., 2014. Clustering for Forensic Analysis. *IMPACT: International Journal of Research in Engineering & Technology (IMPACT: IJRET)*, 2(4), pp. 129-136.
- Khoo, C. S. G. & Poo, D. C. C., 1994. An expert system approach to online catalog subject searching. *Information Processing and Management: an International Journal*, 30(2), pp. 223 -238.
- Kohavi, R., 1995. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *International Joint Conference on Artificial Intelligence (IJCAI)*, Volume 2, pp. 1137-1143 .
- Kruse, W. G. & Heiser, J. G., 2001. *Computer Forensics: Incident Response Essential*. 1st ed. New York: Addison-Wesley Professional.
- Landauer, M. et al., 2018. Dynamic Log File Analysis: An Unsupervised Cluster Evolution Approach For Anomaly Detection. *Computers & Security*, Volume 79, pp. 94-116.
- Leouski, A. V. & Croft, W. B., 1996. *An Evaluation of Techniques for Clustering Search Results*. Amherst, University of Massachusetts at Amherst.
- Leuski, A., 2001. *Evaluating Document Clustering for Interactive Information Retrieval*. New York, ACM, pp. 33-40.
- Leuski, A. & Allan, J., 2000. *Improving Interactive Retrieval by Combining Ranked Lists and Clustering*. France, College de France, pp. 665-681.
- Leuski, A. & Allan, J., 2004. Interactive information retrieval using clustering and spatial proximity. *User Modeling and User-Adapted Interaction*, Volume 14, pp. 259-288.
- Liddy, E. D., 2000. Text Mining. *Bulletin of the American Society for Information Science*, October/November, 27(1), pp. 13-14.

- Li, H. et al., 2018. The Application of Association Analysis in Mobile Phone Forensics System. In: *Intelligence Science II. ICIS 2018. IFIP Advances in Information and Communication Technology Conference Proceedings*. Beijing, China: Springer, Cham, pp. 126-133.
- Lloyd, S. P., 1982. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2), pp. 129 - 137.
- MacQueen, J., 2008. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, Volume 2, pp. 281-297.
- Makhabel, B., Heimann, R., Mishra, P. & Danneman, N., 2017. *R: Mining Spatial, Text, Web, and Social Media Data*. 1st ed. Birmingham, UK: Packt Publishing.
- Manning, C. D. & Schütze, H., 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press.
- Martinc, M., Škrjanec, I., Zupan, K. & Pollak, S., 2018. *PAN 2017: Author Profiling-Gender and Language Variety Prediction*. Dublin, PAN @ CLEF (Conference and Labs of the Evaluation Forum).
- McGuigan, N., 2010. *Text Analytics With Rapidminer Part 2 of 6 - Processing Text*. [Online] Available at: <http://vancouverdata.blogspot.ae/2010/11/text-analytics-with-rapidminer-part-2.html> [Accessed 02 July 2016].
- Mesleh, A., 2007. *Support Vector Machines based Arabic Language Text Classification System: Feature Selection Comparative Study*. Bridgeport, CT, Proceedings of the 2007 International Conference on Systems, Computing Sciences and Software Engineering (SCSS).
- Mesleh, A. M. & Kanaan, G., 2008. *Support vector machine text classification system: Using Ant Colony Optimization based feature subset selection*. Cairo, Egypt, 2008 International Conference on Computer Engineering & Systems.
- Microsoft, 2012. *Excel: How to Find the Percentage of Change Between Values*. [Online] Available at: Ref.: <http://support.microsoft.com/en-ae/help/214078/excel-how-to-find-the-percentage-of-change-between-values> [Accessed 10 Oct 2018].
- Mohammad, A. H., Alwada'n, T. & Al-Momani, O., 2016. Arabic Text Categorization Using Support Vector Machine, Naïve Bayes and Neural Network. *GSTF Journal on Computing (JOC)*, 5(1), pp. 108-115.
- Mohammed, H., Clarke, N. & Li, F., 2016. An Automated Approach for Digital Forensic Analysis of Heterogeneous Big Data. *The Journal of Digital Forensics, Security and Law*, 11(2), pp. 137-152.
- Nemati, H. R. & Barko, C. D., 2003. *Organizational Data Mining: Leveraging Enterprise Data Resources for Optimal Performance*. First ed. PA: Idea Group Publishing.

- Nirkhi, S., Dharaskar, R. V. & Thakare, V. M., 2015. *Authorship Verification of Online Messages for Forensic Investigation*. In: *International Conference on Information Security & Privacy (ICISP2015)*. Nagpur, Elsevier.
- Oh, J., Lee, S. & Lee, S., 2011. *Advanced evidence collection and analysis of web browser activity*. New Orleans, LA, Elsevier Ltd.
- Okolica, J. S., Peterson, G. L. & Mills, R. F., 2007. Using Author Topic to Detect Insider Threats from Email Traffic. *Digital Investigation: The International Journal of Digital Forensics & Incident Response*, 4(3-4), pp. 158-164.
- Olivier, M., 2009. On Metadata Context In Database Forensics. *Digital Investigation*, 5(3-4), pp. 115-123.
- Pallavi, K., NagarjunaReddy, S. & Reddy, D. S. S. S., 2014. Clustering of Documents in Forensic Analysis for Improving Computer Inspection. *International Journal of Advanced Research in Computer and Communication Engineering*, 3(7), pp. 7590-7593.
- Pollitt, M., 1995. *Computer Forensics: an Approach to Evidence in Cyberspace*. Baltimore, MD, Proceedings of the National Information Systems Security Conference.
- Quick, D. & Choo, K.-K. R., 2018. *Big Digital Forensic Data: Volume 1: Data Reduction Framework and Selective Imaging*. 1st ed. Singapore: Springer.
- Quintana, M., Uribe, S., Sánchez, F. & Álvarez, F., 2015. *Recommendation Techniques in Forensic Data Analysis: A New Approach*. London, UK, 6th International Conference on Imaging for Crime Prevention and Detection (ICDP-15).
- Rahman, S. S. u., 2012. *Applied Data Mining*. [Online]
Available at: <http://applieddatamining.blogspot.in/2012/09/k-nn-classification-using-rapidminer.html>
[Accessed 05 08 2018].
- Ramesh, L. S. et al., 2015. Prediction of User Interests for Providing Relevant Information Using Relevance Feedback and Re-ranking. *International Journal of Intelligent Information Technologies*, 11(4), pp. 55-71.
- RapidMiner, 2018. *Operator Reference Guide*. [Online]
Available at: <https://docs.rapidminer.com/latest/studio/operators/>
[Accessed 08 08 2018].
- Rathod, J. N. & Patel, V. R., 2015. Document Clustering in Forensic Analysis: A Review. *International Journal of Data Mining Techniques and Applications*, 4(1), pp. 477-481.
- Reinsel, D., Gantz, J. & Rydning, J., 2018. *The Digitization of the World – From Edge to Core*. [Online]
Available at: <https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-data-age-whitepaper.pdf>
[Accessed 25 01 2019].

- Rijsbergen, C. J. v., 1979. *Information Retrieval*. 2nd ed. MA, USA: Butterworth-Heinemann.
- Robertson, S. E., 1997. The Probability Ranking Principle in IR. In: K. S. Jones & P. Willett, eds. *Readings in Information Retrieval*. San Francisco: Morgan Kaufmann Publishers Inc, pp. 281-286.
- Robertson, S. & Jones, K. S., 1976. Relevance weighting of search terms. *Journal of the American Society for Information Sciences*, 27(3), pp. 129-146.
- Rocchio, J., 1965. *Information Storage and Retrieval: Scientific Report No. ISR-9*. Cambridge, Massachusetts: The National Science Foundation.
- Rocchio, J., 1971. Relevance Feedback in Information Retrieval. In: G. Salton, ed. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Englewood Cliffs, New Jersey: Prentice-Hall Inc, pp. 313-323.
- Roussev, V. & Quates, C., 2012. Content Triage with Similarity Digests: The M57 Case Study. *Digital Investigation*, Volume 9, p. S60–S68.
- Rowe, N. & Garfinkel, S. L., 2012. *Finding Anomalous and Suspicious Files from Directory Metadata on a Large Corpus*. Dublin, 3rd International ICST Conference on Digital Forensics and Cyber Crime.
- Russell, S. & Norvig, P., 2010. *Artificial Intelligence: A Modern Approach*. New Jersey: Pearson Education Inc.
- Ruthven, I. & Lalmas, M., 2003. A survey on the use of relevance feedback for information access systems. *The Knowledge Engineering Review*, 18(2), pp. 95 - 145.
- Saad, M. K., 2010. "Open Source Arabic Language and Text Mining Tools". [Online] Available at: <https://sourceforge.net/projects/ar-text-mining/files/Arabic-Corpora/> [Accessed 02 07 2016].
- Saad, M. K. & Ashour, W., 2010. *OSAC: Open Source Arabic Corpus*. Cyprus, EEECS10 the 6th International Symposium on Electrical and Electronics Engineering and Computer Science.
- Sahoo, D., Bhoi, A. & Balabantaray, R. C., 2018. Hybrid Approach to Abstractive Summarization. *Procedia Computer Science*, Volume 132, pp. 1228-1237.
- Salloum, S. A., Al-Emran, M., Monem, A. A. & Shaalan, K., 2018. Using Text Mining Techniques for Extracting Information from Research Articles. In: K. Shaalan, A. E. Hassanien & M. F. Tolba, eds. *Intelligent Natural Language Processing: Trends and Applications. Studies in Computational Intelligence, vol 740*. Cham: Springer International Publishing, pp. 373-397.
- Salton, G., 1968. *Automatic Information and Retrieval*. 1st ed. NY: McGraw-Hill Inc.
- Salton, G., 1971. *The SMART Retrieval System - Experiments in Automatic Document Processing*, Englewood Cliffs, New Jersey: Prentice-Hall Inc.

- Salton, G. & Buckley, C., 1990. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4), pp. 288-297.
- Salton, G. M., Wong, A. & Yang, C., 1975. A vector space model for automatic indexing. *Communications of the ACM*, Nov., 18(11), pp. 613-620.
- Sardar, T. H. & Ansari, Z., 2018. An Analysis of Mapreduce Efficiency in Document Clustering Using Parallel K-Means Algorithm. *Future Computing and Informatics*, Volume 3, pp. 200-209.
- Saunders, M., Lewis, P. & Thornhill, A., 2012. *Research Methods for Business Students*. 6th ed. Essex: Pearson Education Limited.
- Sawaf, H., Zaplo, J. & Ney, H., 2001. *Statistical Classification Methods for Arabic News Articles*. Toulouse, France, Natural Language Processing in ACL2001.
- Sharma, A., 2018. Cybercrime to Remain an Expensive Foe with More Waging Battle over Email in 2019. *The National*, 31 December, p. 1.
- Shaw, A. & Browne, A., 2013. A Practical and Robust Approach to Coping with Large Volumes of Data Submitted for Digital Forensic Examination. *Digital Investigation*, 10(2), , 10(2), p. 116–128.
- Silva, R. M., Almeida, T. A. & Yamakami, A., 2017. MDLText: An Efficient and Lightweight Text Classifier. *Knowledge-Based Systems*, Volume 118, pp. 152-164.
- Skabar, A., 2017. Clustering Mixed-Attribute Data using Random Walk. *Procedia Computer Science*, Volume 108, p. 988–997.
- Stevens, M. W., 2004. Unification of Relative Time Frames for Digital Forensics. *The International Journal of Digital Forensics & Incident Response*, 1(3), p. 225–239.
- Stewart, J., 2014. *Sifter*. [Online]
Available at: <https://github.com/jonstewart/Sifter>
[Accessed 12 10 2017].
- Syiam, M. M., Fayed, Z. T. & Morgan, M. B. H., 2006. An Intelligent System For Arabic Text Categorization. *International journal of cooperative information systems*, 6(1), pp. 1-19.
- Takahashi, D. et al., 2010. IEEE 802.11 User Fingerprinting and Its Applications for Intrusion Detection. *Computers & Mathematics with Applications*, 60(2), pp. 307-318.
- Tanaseichuk, O. et al., 2015. An Efficient Hierarchical Clustering Algorithm for Large Datasets. *Austin Journal of Proteomics, Bioinformatics & Genomics*, 2(1), pp. 1-6.
- Tan, P.-N., Steinbach, M. & Kumar, V., 2005. *Introduction to Data Mining*. First ed. Boston: Addison-Wesley Longman Publishing Co., Inc.

- Thilagavathi, G. & Anitha, J., 2014. Document Clustering in Forensic Investigation by Hybrid Approach. *International Journal of Computer Applications (0975 – 8887)*, 91(3).
- Tsimperidis, I. A. A. & Karakos, A., 2018. Keystroke Dynamics Features for Gender Recognition. *Digital Investigation*, Volume 24, pp. 4-10.
- Tsimperidis, I., Katos, V. & Clarke, N. L., 2015. Language - Independent Gender Identification Through Keystroke Analysis. *Information Management & Computer Security*, 23(3), pp. 286-301.
- Turnbull, D. & Berryman, J., 2016. *Relevant Search: With applications for Solr and Elasticsearch*. 1st ed. Shelter Island, NY: Manning Publications.
- Uteuov, A. & Kalyuzhnaya, A., 2018. Combined Document Embedding and Hierarchical Topic Model for Social Media Texts Analysis. *Procedia Computer Science*, Volume 136, p. 293–303.
- Vallejo-Huanga, D., Morillo, P. & Ferri, C., 2017. *Semi-Supervised Clustering Algorithms for Grouping Scientific Articles*. In: *International Conference on Computational Science, ICCS 2017*. Zurich, Elsevier B.V.
- Varma, S., Walls, R. J., Lynn, B. & Levine, B. N., 2014. *Efficient Smart Phone Forensics Based on Relevance Feedback*. Scottsdale, Arizona, Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices.
- Venturini, L., Garza, P. & Apiletti, D., 2016. *BAC: a Bagged Associative Classifier for Big Data Frameworks*. In: *East European Conference on Advances in Databases and Information Systems*. Cham, Springer.
- Vidhya, B. & Vijayanthi, P. R., 2014. Enhancing Digital Forensic Analysis through Document Clustering. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 3(1), pp. 127-130.
- Wagner, J. et al., 2017. Carving Database Storage to Detect and Trace Security Breaches. *Digital Investigation*, 22(Supplement), pp. S127-S136.
- Walnycky, D. et al., 2015. *Network and device forensic analysis of Android social-messaging applications*. Philadelphia, PA, Elsevier Ltd, p. S77–S84.
- Wang, D., Han, B. & Huang, M., 2012. *Application of Fuzzy C-Means Clustering Algorithm Based on Particle Swarm Optimization in Computer Forensics*. China, Elsevier B.V.
- Wiles, J., 2011. *TechnoSecurity's Guide to E-Discovery and Digital Forensics: A Comprehensive Handbook*. 1st ed. Burlington: Syngress Publishing Inc.
- Witten, I. H., Frank, E., Pal, C. J. & Hall, M. A., 2016. *Data Mining*. 4th ed. Burlington, MA: Morgan Kaufmann Publishers.
- Zaeem, R. N., Manoharan, M., Manoharan, M. & Yang, Y., 2017. Modeling and Analysis of Identity Threat Behaviors Through Text Mining of Identity Theft Stories. *Computers & Security*, Volume 65, pp. 50-63.


Zamir, O. & Etzioni, O., 1998. *Web Document Clustering: A Feasibility Demonstration*. New York, ACM, pp. 46-54.

Zamir, O. & Etzioni, O., 1999. Grouper: A Dynamic Clustering Interface to Web Search Results. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 31(11-16), pp. 1361-1374.

Zrigui, M., Ayadi, R., Mars, M. & Maraoui, M., 2012. Arabic Text Classification Framework Based on Latent Dirichlet Allocation. *Journal of Computing and Information Technology*, 20(2), p. 125–140.

Appendix

1. Appendix – Research Authorization Letter



The British University
in Dubai

Subject: Letter of Authorization to Conduct Research at EEEEC

Dear Emirates Electronic Evidence Center Director,

I am a student in The British University in Dubai (BUiD) undertaking a research study in the PhD in Computer Science program. I would like to kindly request your permission to conduct my research entitled “**Relevance Feedback Optimization for Digital Forensic Investigations**” at your Center.

The research will consist of interviews with digital forensic examiners who will be asked to test the developed tool. In order to showcase the effectiveness of this research, it will be using real world data from selected closed cases from your Center.

This letter will serve as authorization for the researcher to conduct abovementioned research project at the Emirates Electronic Evidence Center (EEEC) located in Abu Dhabi Police’s CID Directorate. Additionally, the interviews will take place at your premises or another appropriate location and when the examiner’s time allows it.

The EEEEC acknowledges that it has reviewed the request presented by the researcher, accepts and authorizes the research project to proceed. The research project may be implemented at the Center premises, and the results will be made available for your preview prior to any publication of reports.

Sincerely,
The Researcher
Hanadi Al Suwaidi

Date 26.6.18

Major/ Tariq Al Zaabi (EEEC Director)
Name & Title of Authorized Signatory

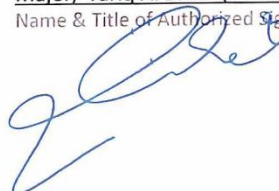


Figure 1.1: Research Authorization Letter

2. Appendix – Search & Visualization Experiment

2.1. Setup with dtSearch (Test 1)

The files in the dataset were indexed using dtSearch Indexer (dtSearch Desktop version 7.83).

The tool stored the location of each word from the collection to be easily searched using the dtSearch Desktop. The indexed results are searchable by the user querying the collection for specific terms (Figure 2.1), which can be a simple word, Boolean searches, or regular expressions.

Participant 1 was given an overview of the dtSearch features. Then they were provided with 3 main questions (Appendix 2.3 – Test 1) and were asked to answer them. The Search History from Test 1 is shown in (Figure 2.2).

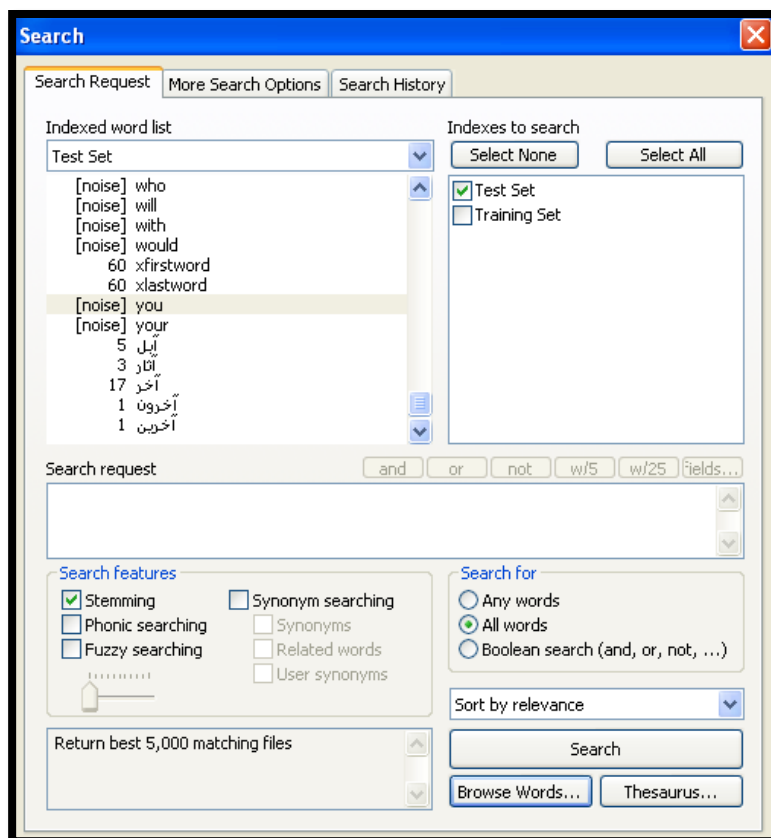


Figure 2.1: dtSearch Desktop Search Feature

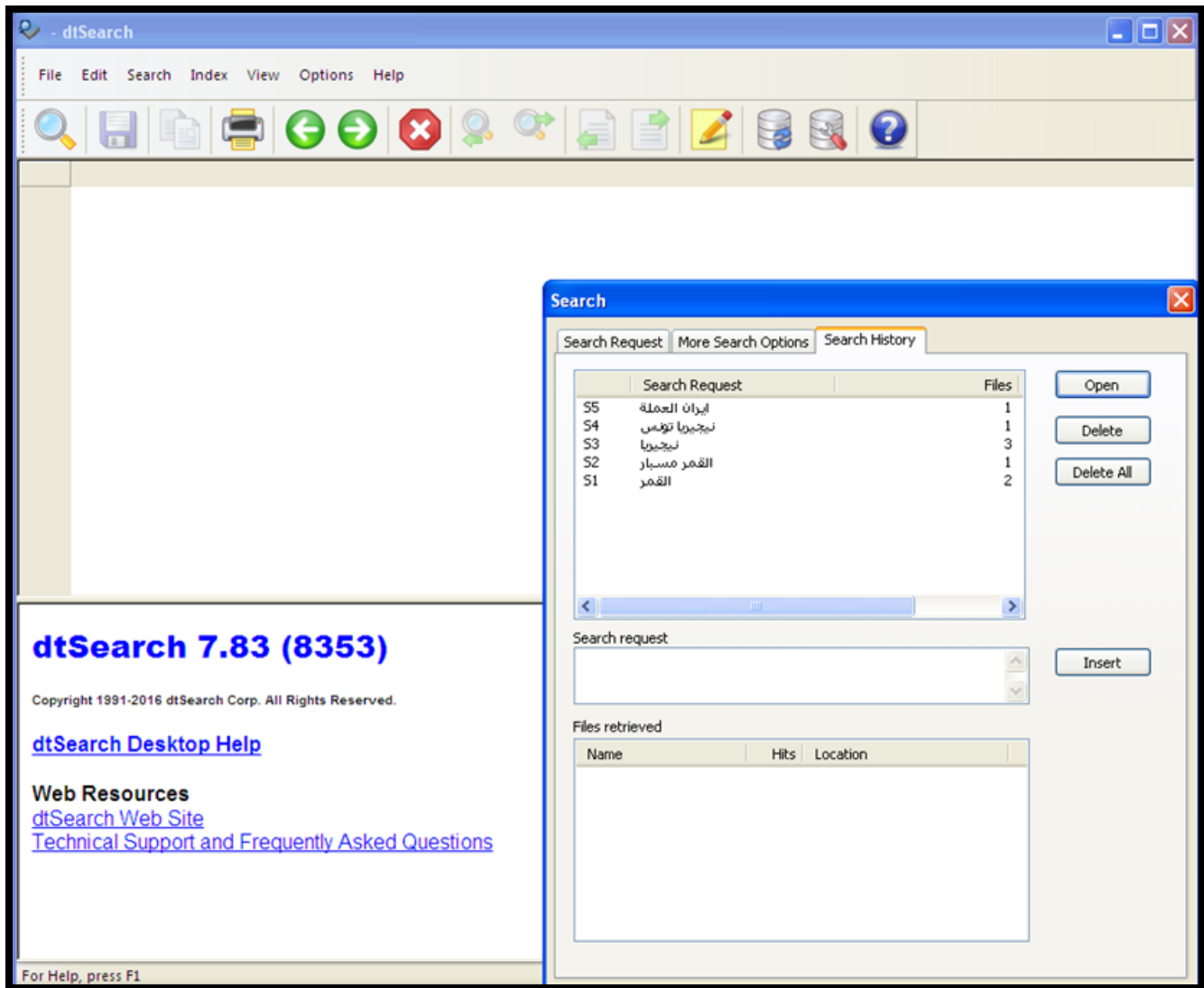


Figure 2.2: dtSearch Desktop Search History (Test 1)

2.2. Setup with RapidMiner (Test 2)

RapidMiner Studio version 7.1 was used to cluster the file collections. This section gives an overview of the main processes that were taken to achieve that.

Step 1: Process Documents from Files (Figure 2.3)

1. Transform Cases (as some articles might contain English words, this operator was used to transform all letters to lower case)
2. Tokenize (in order to tokenize words)
3. Filter Tokens (filtered by word length, which was set to 3 or more)
4. Filter Stopwords (filtered out Arabic Stopwords)

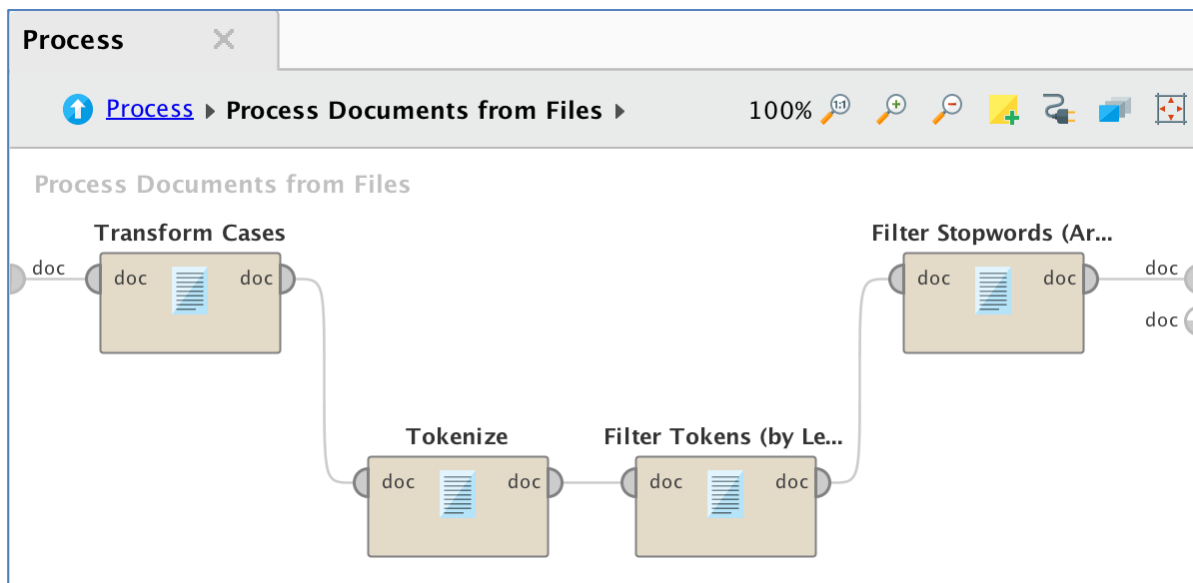


Figure 2.3: Processing Documents Operators

Step 2: Configure the “Process Documents from Files” Process

As this process was used to generate word vectors from the sample, the Term Frequency–Inverse Document Frequency (TF-IDF) schema was used to create these vectors. The TF-IDF reflects how important a term in a document is in relation to the collection, which in this study was essential to figure out a term’s significance in relation to the category they’re placed in.

Step 3: Clustering

The processed documents from the previous steps would then be clustered using the K-Means algorithm; in order to group similar and dissimilar files into different clusters (Figure 2.4).

During the tests run on the Training Set, the results from different K values were evaluated by the researcher; the results were too condensed when the K value was 4 or less, and too distributed when it went over 10. Thus, the best value for K was 6, and further clustering of those generated clusters (to create the sub clusters) was grouped using a K value of 3.

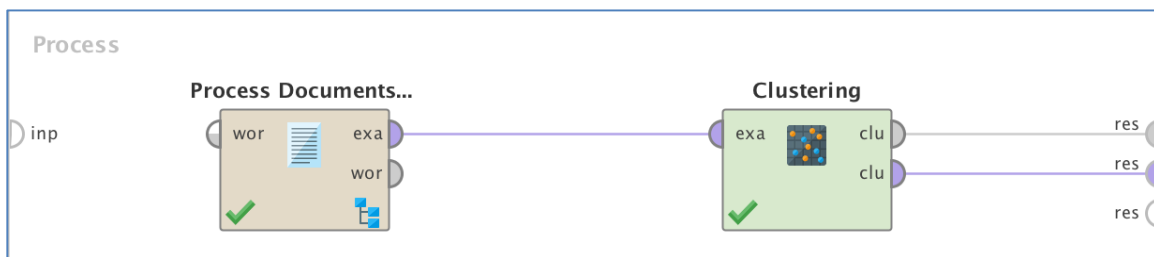


Figure 2.4: Clustering of Processed Documents

Furthermore, the numerical measure type was set to the Euclidean Distance. Both Cosine Similarity and Euclidean Distances were tested during the test runs on the Training Set. The resulting clusters were similar, and while the Cosine measured how similar two documents were,

the Euclidean measured the magnitude of their differences. At the end, the Euclidean measure was used on the dataset.

In order to provide the participant with a user friendly interface to navigate the generated word clouds, a simple HTML website was created and presented to the user (Figure 2.5). The participant was provided with 3 main questions for Experiment 2 (Appendix 2.3 – Test 2) and asked to answer them, after they were given an overview of how to navigate the website, move from the Main Clusters (Figure 2.6) to their Sub-Clusters (Figures 2.7), and how to preview file contents from each cluster. The word clouds were created from the generated clusters as follows:

1. The clustering process resulted in 6 different clusters (**Main Clusters: 0 - 5**), which were further used as a source for the next level of clustering.
2. The Main Clusters' files were further clustered into 3 different clusters (**Sub-Clusters: 0 - 2**) using the same settings as the first round of clustering.
3. The 10 most frequent words from each of the clusters, Main & Sub-Clusters, were used to generate word clouds through the website (<https://wordsift.org/>).

The screenshot shows a web application interface with a dark navigation bar at the top containing the links "Home", "Groups", and "Subgroups", and a hamburger menu icon on the right. Below the navigation bar, the main content area is white and contains the following text:

Hello,
Thank you for participating in this experiment.

In this test, word collages of file collections were created to assist in answering the given questions, please click on the desired group to drill down further into its subgroups, then examine the files to search for the correct answer. Enjoy!

Questions

Please answer the following questions based on the provided information and note down both the answer and filename.

Question 1: How many barrels of oil did the Emirates produce daily in the year that the International Renewable Energy Agency started in the UAE?

Question 2: What was the sample size for the people that participated in the largest clinical trial in the world to produce a new vaccine for the virus that causes AIDS?

Question 3: In which stadium did the final football match between the Algerian and Egyptian national teams take place for World Cup qualifiers for South Africa in 2010?

Figure 2.5: Main Page to Navigate to Groups (Main Clusters)

Home Groups Subgroups

Groups

Group 1



المباراة

القاهرة
المنتخب الجزائري

المصرية
الجزائري

المصري

مصر

[More like this —>](#)

Group 2



اشتريت

دبي

الامارات

نخيل

**Figure 2.6: Each Group (Main Cluster)
Links to its own Subgroups (Sub Clusters)**

Home Groups Subgroups

Subgroup 1

Subgroups

[Back to Groups](#)

Group 1
Subgroup 1

بيروت

البرازيل ^{بطل}

البنانية

الفرنكوفونية

التجارة

ريشيد النامية

متر ^{متر}

Files: [30](#), [48](#), [52](#), [58](#)

Subgroup 2

الجماهير

الدقيقة الشوط

التذاكر

التي

المباراة

الثاني

الحضري

Files: [13](#), [17](#), [35](#)

Subgroup 3

القاهرة

البلدين

Figure 2.7: Subgroups (Sub Clusters)

2.3. Test Questions

Dear Participant,

Thank you for participating in the following experiments. The files under investigation are text based news articles. Please use dtSearch to answer the first set of questions by using your own search terms and reviewing the generated hits. Note down each answer, all the search terms you've used, and the filename you found the answer in.

For the second set of questions, please use the provided website to navigate the word clouds that were created to assist you in answering the given questions. Please click on the desired group to drill down further into its subgroups, and then review the files to search for the correct answer. Note down each answer, and the filename you found the answer in.

Thank you!

Test 1 - dtSearch Questions

Q1: What is the name of the probe that was used by NASA during their search for water in the Moon?

Q2: Who scored the equalizer goal that tied the score between Tunisia and Nigeria during the last minute of their match?

Q3: What was the currency that Iran decided to price its oil in instead of the Dollars months before the recent crisis?

Q4: After taking a quick look at the files content in this collection, how many news categories do you think were distributed over the sample?

Test 2 – Word Collages Questions

Q1: How many barrels of oil did the Emirates produce daily in the year that the International Renewable Energy Agency started in the UAE?

Q2: What was the sample size for the people who participated in the largest clinical trial in the world to produce a new vaccine for the virus that causes AIDS?

Q3: In which stadium did the final football match between the Algerian and Egyptian national teams take place for World Cup qualifiers for South Africa in 2010?

Q4: How many categories do you think were distributed over the sample?

2.4. File Clusters & Word Clouds

The Main Clusters Centroids (Top 20 Terms) are shown in (Table 2.1), while the Sub-Clusters Centroids (Top 10 Terms) are shown in (Tables 2.2-2.7).

Freq. Order	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
1	القاهرة	دبي	القمة	الدقيقة	الدولار	الانترنت
2	الجزائر	اشترت	المناخ	مانتسستر	النفط	المعلومات
3	المصري	العالمية	الكربون	يونائتد	المئة	البرامج
4	المنتخب	حصة	كوبنهاجن	الحارس	العملات	الشمسية
5	المباراة	شركة	اتفاق	الشوط	العراق	التجربة
6	الجزائري	ديون	قمة	الكرة	حقول	الهواتف
7	مصر	بورصة	المناخي	الحضري	التقرير	الإنترنت
8	المصرية	الامارات	مصدر	المباراة	سعر	الجديد
9	التي	نخيل	المفاوضات	روني	الحقول	عملية
10	البلدين	المئة	الفضاء	الهدف	حقل	استخدام
11	كأس	الان	الحراري	الجزائر	الخليج	الدراسة
12	الجزائرية	الازمة	انعقاد	تريكة	العراقية	الإصابة
13	اللاعبين	المالية	المناخية	هدف	الاقتصاد	الأشخاص
14	الجماهير	الشركة	المؤتمر	المنتخب	العام	الساعة
15	استاد	مليار	الطاقة	تشيلسي	النفطية	السرعة
16	بالحجارة	حكومة	التغيرات	الثاني	التضخم	الباحثون
17	الاتحاد	بنك	الدراسة	ميلان	الشركات	العلماء
18	الخارجية	صندوق	الرئيس	مصر	الفقيرة	فيروس
19	القدم	لحكومة	سطح	التعادل	الناتج	نتائج
20	مباراة	الاوربية	الدول	الإيطالي	دول	الهاتف

Table 2.1: Top 20 Terms from Main Clusters

Freq. Order	Cluster 0	Cluster 1	Cluster 2
1	متر	التي	المصرية
2	التجارة	الجماهير	احد
3	رشيد	الدقيقة	القاهرة
4	اللبنانية	الثاني	الفيفا
5	البرازيل	الشوط	الجزائري
6	النامية	التذاكر	البلدين
7	بيروت	المباراة	الحادثة
8	الفرنكوفونية	الحضري	الخارجية
9	بطل	الحارس	الجزائر
10	الدول	استاد	الجزائرية

Table 2.2: Top 10 Terms from Sub Clusters of (Main Cluster 0)

Freq. Order	Cluster 0	Cluster 1	Cluster 2
1	اشترت	الان	السوق
2	المئة	استفادت	ازمة
3	حصة	الاسماء	دول
4	بنسبة	الدين	مشاكل
5	بنك	نقدا	الدول
6	صندوق	وجهة	النفطية
7	مقابل	الرواج	تصور
8	مليون	اخرى	ولا
9	نخيل	الانتماء	بالمليارات
10	صالح	التسوق	الدعم

Table 2.3: Top 10 Terms from Sub Clusters of (Main Cluster 1)

Freq. Order	Cluster 0	Cluster 1	Cluster 2
1	القمر	الازمة	القمة
2	الأطفال	المدينة	مقر
3	سيرنان	الطاقة	الوزراء
4	التغير	الان	اتفاق
5	الدراسة	مصدر	بشأن
6	الفضاء	التعافي	المشاركة
7	الدكتور	الركود	الوفود
8	الوراثية	الامارات	راسموسين
9	المواقع	اوباما	رئيس
10	المناخي	النمو	انعقاد

Table 2.4: Top 10 Terms from Sub Clusters of (Main Cluster 2)

Freq. Order	Cluster 0	Cluster 1	Cluster 2
1	متر	التي	المصرية
2	التجارة	الجماهير	احد
3	رشيد	الدقيقة	القاهرة
4	اللبنانية	الثاني	الفيفا
5	البرازيل	الشوط	الجزائري
6	النامية	التذاكر	البلدين
7	بيروت	المباراة	الحادثة
8	الفرنكوفونية	الحضري	الخارجية
9	بطل	الحارس	الجزائر
10	الدول	استاد	الجزائرية

Table 2.5: Top 10 Terms from Sub Clusters of (Main Cluster 3)

Freq. Order	Cluster 0	Cluster 1	Cluster 2
1	الدولار	الأحياء	المئة
2	أسعار	الفقيرة	نمو
3	صرف	العراق	العام
4	العملات	حقل	الاسترليني
5	الخضراء	نسمة	الناتج
6	بالدولار	حقول	الاقتصاد
7	الأمريكي	العراقية	البريطاني
8	الورقة	الحقول	اسواق
9	سعر	الشركات	الاجمالي
10	الإنديبننت	الجولة	المحلي

Table 2.6: Top 10 Terms from Sub Clusters of (Main Cluster 4)

Freq. Order	Cluster 0	Cluster 1	Cluster 2
1	الانترنت	الخطة	الشمسية
2	البرامج	القمر	اللحوم
3	الاستطلاع	ناسا	الاستفتاء
4	فون	الارتطام	الخنازير
5	الوصول	لدينا	hiv
6	اسرة	الغبار	والحمامات
7	كيم	المهمة	التقنية
8	الجنوبية	اللجنة	أيفون
9	كوريا	أمريكي	إنفلونزا
10	الكمبيوتر	الطيف	الأطفال

Table 2.7: Top 10 Terms from Sub Clusters of (Main Cluster 5)

The following (Table 2.8) displays the distribution of files over the Main and Sub-Clusters.



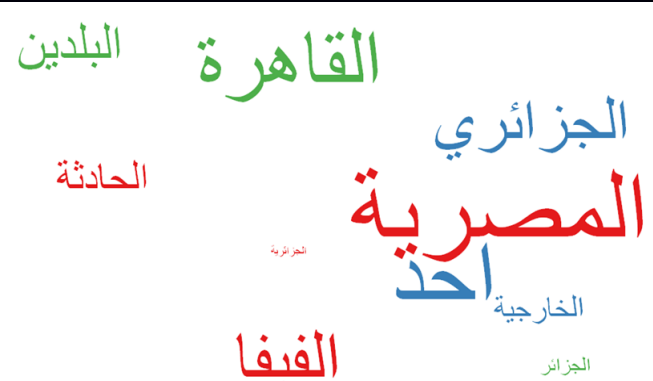
Table 2.8: Sub Clusters' Files & Word Clouds			
Main Cluster #	Sub Cluster #	Filename (.txt)	Sub Cluster (Word Collage)
0	0	30	
		48	
		52	
		58	
	1	13	
		17	
		35	
		26	
	2	37	
		49	
		50	

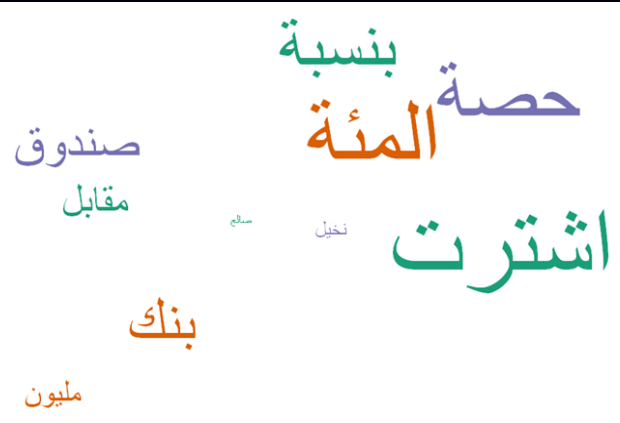


Table 2.8: Sub Clusters' Files & Word Clouds			
Main Cluster #	Sub Cluster #	Filename (.txt)	Sub Cluster (Word Collage)
		53	
1	0	3	
		4	
		15	
		33	
	1	20	
	2	27	
29			

Table 2.8: Sub Clusters' Files & Word Clouds			
Main Cluster #	Sub Cluster #	Filename (.txt)	Sub Cluster (Word Collage)
2	0	1	<p>الأطفال</p> <p>الوراثية سيرنان المواقع</p> <p>الفضاء الدراسة القمر</p> <p>التغير الدكتور</p>
		2	
		9	
		32	
	1	16	<p>الان مصدر</p> <p>الازمة</p> <p>الطاقة الركود</p> <p>التعافي المدينة</p> <p>الامارات</p>
		23	
		54	
	2	10	<p>راسموسين</p> <p>اتفاق القمة مقر رئيس</p> <p>المشاركة الوزراء</p> <p>بشأن الوفود</p>
		41	
	3	0	44
51			
57			

Table 2.8: Sub Clusters' Files & Word Clouds			
Main Cluster #	Sub Cluster #	Filename (.txt)	Sub Cluster (Word Collage)
	1	59	<p>المجموعة دور</p> <p>تشلسي</p> <p>ارضه</p> <p>ولا</p> <p>اهداف</p> <p>بايرن</p> <p>بور دو</p>
	2	22	<p>مصر</p> <p>الجزائري</p> <p>الحضري</p> <p>تريكة</p> <p>الجزائر</p> <p>المصري</p> <p>عبد</p> <p>نيجيريا</p>
		31	
		56	
		60	
4	0	12	<p>بالدولار</p> <p>صرف</p> <p>الدولار</p> <p>الخضراء</p> <p>العملات</p> <p>أسعار</p> <p>الأمريكي</p>
		14	
		42	
	1	24	<p>العراقية</p> <p>حقول</p> <p>الفقيرة</p> <p>نسمة</p> <p>حقول</p> <p>الأحياء</p> <p>العراق</p>
		28	
		40	

Table 2.8: Sub Clusters' Files & Word Clouds			
Main Cluster #	Sub Cluster #	Filename (.txt)	Sub Cluster (Word Collage)
	2	43	
		6	<p>الناتج اسواق نمو</p> <p>الاجمالي المنة</p> <p>الاقتصاد</p> <p>البريطاني</p> <p>الاسترليني العام</p>
		11	
		45	
		46	
		55	
5	0	8	<p>الاستطلاع الانترنت</p> <p>الجنوبية</p> <p>الوصول كوريا</p> <p>اسرة البرامج</p> <p>فون كيم</p>
		18	
		34	
	1	19	<p>لدينا ناسا المهمة</p> <p>القمر</p> <p>الخطة</p> <p>الغبار</p> <p>اللجنة الارتمام امريكي</p>
		25	

Table 2.8: Sub Clusters' Files & Word Clouds			
Main Cluster #	Sub Cluster #	Filename (.txt)	Sub Cluster (Word Collage)
	2	5	
		7	
		21	
		36	
		38	
		39	
		47	

Table 2.8: Sub Clusters' Files & Word Clouds

3. Appendix – PoC Filters & Operators

This Appendix chapter provides details on the used RapidMiner Filters and Operators.

3.1. RapidMiner Filters (Preprocessing)

RapidMiner provides a Text Processing extension that could be found at the RapidMiner Marketplace. This extension provided helpful operators for statistical text analysis and Natural Language Processing (NLP).

The Text Processing extension was useful, as it supported several text formats such as plain text, PDF, and other data sources. It also provided handy filters for tokenization and Stopword for Arabic text, which were necessary for preprocessing of text before running it through an algorithm for text analysis. The following RapidMiner (2018) filters were used for the different steps of the proposed framework.

Tokenize: which splits the document text into a sequence of tokens.

- Extensions
 - Text Processing
 - Tokenization
 - **Tokenize**

Stopword (Arabic): uses the built-in Stopword list to remove from a document every token that equals a Stopword from the built-in list. In order for this filter to work correctly, every word needs to have been tokenized beforehand. Consequently, this filter appears after the Tokenize operator and not before it.

- Extensions
 - Text Processing
 - Filtering
 - **Filter Stopwords (Arabic)**

3.2. RapidMiner Operators

The following RapidMiner (2018) operators were used for the different steps of the proposed framework.

3.2.1. Step 1 (Clustering)

Process Documents from Files: this operator uses a text collection stored in multiple files to generate word vectors.

- Extensions
 - Text Processing
 - **Process Documents from Files**

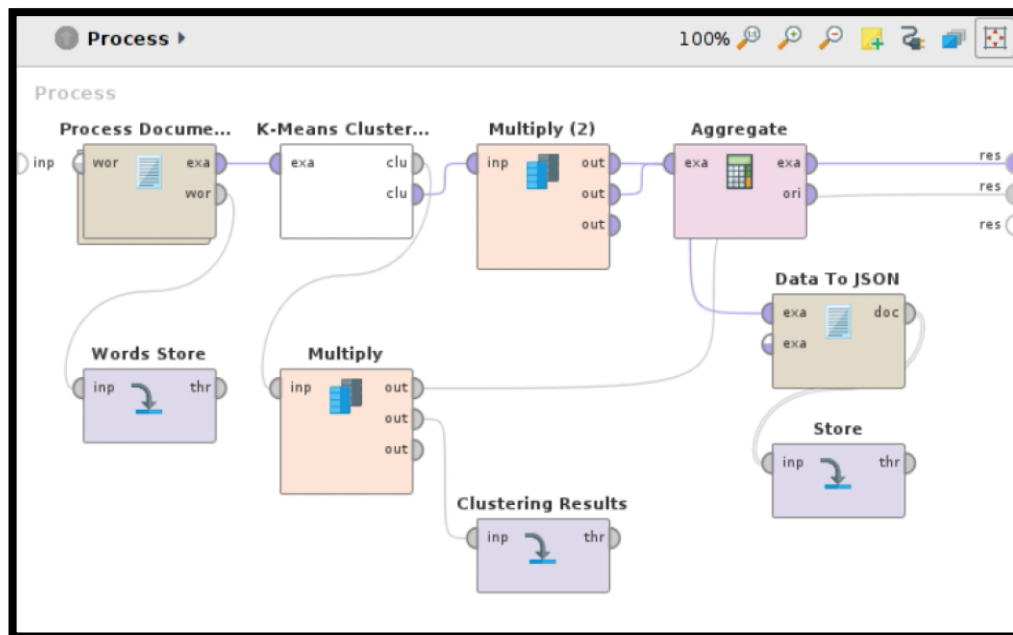


Figure 3.1: Document Clustering Process in RapidMiner Studio

K-Means Clustering: this operator uses the k-means algorithm to perform clustering, by grouping similar Examples together. K-means clustering is an unsupervised learning algorithm that can be used on unlabelled data.

- Modeling
 - Segmentation
 - **K-Means**

Aggregate: this operator uses an input ExampleSet to create a new ExampleSet that shows the results of the selected aggregation functions.

- Blending
 - Table
 - Grouping
 - **Aggregate**

Data to JSON: this operator transforms an ExampleSet into JSON documents, where each Example is converted into a single JSON array containing all Examples, or into a separate document. It also unflattens nested structures which are represented as Attribute names.

- Extensions
 - Text Processing
 - Utility
 - **Data to JSON**

Store: this operator stores an IO Object in the data repository and can be used by other processes by using the Retrieve operator.

- Data Access

- **Store**

3.2.2. Step 2 (Classification)

Create Document: this operator creates a document that contains the text that was given as parameter.

- Extensions
 - Text Processing
 - **Create Document**

De-Pivot: this operator converts examples of the selected attributes (which usually measure the same characteristic) into examples of a single attribute.

- Blending
 - Table
 - Rotation
 - **De-Pivot**

Retrieve: this operator accesses information that is stored in the Repository and loads them into the process.

- Data Access
 - **Retrieve**

Set Role: this operator can be used to change the role of Attributes.

- Blending

- Attributes
 - Names & Roles
 - **Set Role**

Nominal to Text: this operator can be used to map all values of the selected nominal attributes to corresponding string values and to convert all nominal attributes into string attributes.

- Blending
 - Attributes
 - Types
 - **Nominal to Text**

Join: this operator can use one or more Attributes of the input ExampleSets in order to join two ExampleSets.

- Blending
 - Attributes
 - Joins
 - **Join (Concurrncy)**

Rename: this operator can be used for renaming Attributes of the input ExampleSets.

- Blending
 - Attributes
 - Names & Roles
 - **Rename**

Select Attributes: this operator can be used to select a subset of Attributes or to remove Attributes of an ExampleSet.

- Blending
 - Attributes
 - Selection
 - **Select Attributes**

Process Documents from Data: this operator uses string attributes to generate word vectors.

- Extensions
 - Text Processing
 - **Process Documents from Data**

Cross Validation: this operator is used to estimate the performance accuracy of a model (which had learned by a particular learning operator)

- Validation
 - **Cross Validation**

k-NN: this operator is used for classification or regression to generate a k-Nearest Neighbor (k-NN) model based on comparing an unknown Example with the k training Examples (these are the unknown Example's nearest neighbors).

- Modeling
 - Predictive
 - Lazy
 - **K-NN**

Apply Model: usually a model is trained on an ExampleSet using another Operator (i.e. learning algorithm), this model is then applied on an ExampleSet.

- Scoring
 - **Apply Model**

Performance (Classification): this operator generates a list of performance values of the classification task, which explains the statistical performance evaluation of the classification tasks.

- Validation
 - Performance
 - Predictive
 - **Performance (Classification)**

3.2.3. Step 3 (Apply Model)

Remove Duplicates: this operator compares all Examples with each other based on specific attributes and removes duplicate Examples from an ExampleSet.

- Cleansing
 - Duplicates
 - **Remove Duplicates**

Loop Files: this operator can be used to execute the inner process tasks on every selected file and to select and filter files of a directory.

- Utility

- Process Control
 - Loops
 - **Loop Files**

3.2.4. Other Operators

JSON to Data: this operator flattens nested objects and transforms a collection of JSON documents into an ExampleSet, so that each JSON document corresponds to one Example.

- Extensions
 - Text Processing
 - Utility
 - **JSON to Data**

Filter Examples: this operator is used to select the Examples of an ExampleSet that are kept or removed based on user defined conditions.

- Blending
 - Examples
 - Filter
 - **Filter Examples**

Filter Example Range: this operator is used to select the Examples of an ExampleSet that are kept or removed based on Examples within a specified index range.

- Blending
 - Examples
 - Filter

- **Filter Example Range**

Transpose: this operator is used to transpose the input ExampleSet. For example, the current columns become rows of the output ExampleSet and current rows become columns of the output ExampleSet.

- Blending

- Table

- Rotation

- **Transpose**

Sort: this operator sorts the input ExampleSet (based on a single attribute) in ascending or descending order.

- Blending

- Examples

- Sort

- **Sort**

Replace: this operator can be used to select attributes and to specify regular expressions that will match and replace parts of the values of selected nominal attributes that were entered by the users.

- Blending

- Values

- **Replace**

Generate Attributes: this operator can use mathematical expressions to create user defined Attributes taken from the input ExampleSet.

- Blending
 - Attributes
 - Generation
 - **Generate Attributes**

WordList to Data: this operator can use a word list to build a dataset. The dataset contain information such as a row for each word with attributes for the word and the number of labeled documents where it occurred.

- Extensions
 - Text Processing
 - Utility
 - **WordList to Data**

Generate Data by User Specification: this operator derives attributes and their types and roles from the parameters' given expressions in order to generate an ExampleSet that contains exactly one Example.

- Utility
 - Random Data Generation
 - **Generate Data by User Specification**

Append: this operator adds all Examples into a combined set in order to merge ExampleSet from two or more ExampleSets, keeping in mind that the same attribute signature (i.e. the same

number of attributes, and the same names and roles of attributes) should be in the input

ExampleSets.

- Blending
- Table
- Joins
- **Append**

4. Appendix – Detailed Research Results

The detailed research results are shared in this Appendix chapter. The first part shows the individual and combined results for each of the participants, while the second part shows the combined interview transcripts, survey and questionnaire results.

4.1. Detailed PoC Experiment Results

This section shares the participant’s results starting with Encase results then Probo results, respectively for each participant.

4.1.1. Participant 1 Results

Dataset Group (1)	Encase Participant 1	
	Relevant (User)	Not Relevant (User)
Relevant (Correct: 40)	39	
Not Relevant (Correct: 32)		33
Total	40	32
Error Rate = (FP+FN)/ (Total Number of Items) In this example: (1+2)/72 = 0.042	0.042	0.042
Error Rate Avg. This indicates the Total Errors Avg. In this example: (0.042+0.042)/2 = 0.042	4%	
True Positive (TP): Indicates the number of items selected correctly as relevant.	38	31
True Negative (TN): Indicates the number of items selected correctly as relevant.	31	38
False Positive (FP): Indicates the items incorrectly selected as relevant.	1	2
False Negative (FN):	2	1

Indicates the number of items incorrectly selected as non-relevant.		
Precision = TP/(TP+FP)	$38/(38+1) = 0.974$	$33/(33+2) = 0.939$
Recall = TP/(TP+FN)	$38/(38+2) = 0.95$	$33/(33+1) = 0.969$
F-Measure = (2*Precision*Recall)/(Precision+Recall)	0.962	0.954
Avg. Accuracy = 1 - Avg. Error Rate	1 - 0.042 = 96%	

Table 4.1: Participant 1 Experiment Results (Encase)

Dataset Group (2)	Probo Participant 1	
	Relevant (User)	Not Relevant (User)
Relevant (Correct: 40)	33	
Not Relevant (Correct: 32)		39
Total	39	33
Error Rate = (FP+FN)/ (Total Number of Items)	0.1111	0.1111
Error Rate Avg. This indicates the Total Errors' Avg.	11%	
True Positive (TP): Indicates the number of items selected correctly as relevant.	32	32
True Negative (TN): Indicates the number of items selected correctly as relevant.	32	32
False Positive (FP): Indicates the items incorrectly selected as relevant.	1	7
False Negative (FN): Indicates the number of items incorrectly selected as non-relevant.	7	1
Precision: TP/(TP+FP)	$32/(32+1) = 0.97$	$32/(32+7) = 0.82$
Recall: TP/(TP+FN)	$32/(32+7) = 0.82$	$32/(32+1) = 0.97$
F-Measure: (2*Precision*Recall)/(Precision+Recall)	0.888	0.888

Avg. Accuracy = 1 - Avg. Error Rate	89%
--	-----

Table 4.2: Participant 1 Experiment Results (Probo)

4.1.2. Participant 2 Results

Dataset Group (2)	Encase Participant 2	
	Relevant (User)	Not Relevant (User)
Relevant (Correct: 39)	17	
Not Relevant (Correct: 33)		52
Total	39	33
Error Rate = (FP+FN)/ (Total Number of Items)	0.306	0.306
Error Rate Avg. This indicates the Total Errors' Avg.	31%	
True Positive (TP): Indicates the number of items selected correctly as relevant.	17	33
True Negative (TN): Indicates the number of items selected correctly as relevant.	33	17
False Positive (FP): Indicates the items incorrectly selected as relevant.	0	22
False Negative (FN): Indicates the number of items incorrectly selected as non-relevant.	22	0
Precision: TP/(TP+FP)	$17/(17+0) = 1$	$33/(33+22) = 0.6$
Recall: TP/(TP+FN)	$17/(17+22) = 0.436$	$33/(33+0) = 1$
F-Measure: (2*Precision*Recall)/(Precision+Recall)	0.607	0.75
Avg. Accuracy = 1 - Avg. Error Rate	1 - 0.31 = 69%	

Table 4.3: Participant 2 Experiment Results (Encase)

Dataset Group (1)	Probo Participant 2	
	Relevant (User)	Not Relevant (User)

Relevant (Correct: 40)	39	
Not Relevant (Correct: 32)		33
Total	40	32
Error Rate = (FP+FN)/ (Total Number of Items)	0.0139	0.0139
Error Rate Avg. This indicates the Total Errors' Avg.	0.0139	
True Positive (TP): Indicates the number of items selected correctly as relevant.	39	32
True Negative (TN): Indicates the number of items selected correctly as relevant.	32	39
False Positive (FP): Indicates the items incorrectly selected as relevant.	0	1
False Negative (FN): Indicates the number of items incorrectly selected as non-relevant.	1	0
Precision: TP/(TP+FP)	$39/(39+0) = 1$	$32/(32+1) = 0.97$
Recall: TP/(TP+FN)	$39/(39+1) = 0.98$	$32/(32+0) = 1$
F-Measure: (2*Precision*Recall)/(Precision+Recall)	0.987	0.985
Avg. Accuracy = 1 - Avg. Error Rate	1 - 0.0139 = 99%	

Table 4.4: Participant 2 Experiment Results (Probo)

4.1.3. Participant 3 Results

Dataset Group (1)	Encase Participant 3	
	Relevant (User)	Not Relevant (User)
Relevant (Correct: 40)	40	
Not Relevant (Correct: 32)		32
Total	40	32
Error Rate = (FP+FN)/ (Total Number of Items)	0	0
Error Rate Avg. This indicates the Total Errors' Avg.	0	

True Positive (TP): Indicates the number of items selected correctly as relevant.	40	32
True Negative (TN): Indicates the number of items selected correctly as relevant.	32	40
False Positive (FP): Indicates the items incorrectly selected as relevant.	0	0
False Negative (FN): Indicates the number of items incorrectly selected as non-relevant.	0	0
Precision: TP/(TP+FP)	$40/(40+0) = 1$	$32/(32+0) = 1$
Recall: TP/(TP+FN)	$40/(40+0) = 1$	$32/(32+0) = 1$
F-Measure: (2*Precision*Recall)/(Precision+Recall)	1	1
Avg. Accuracy = 1 - Avg. Error Rate	1 - 0 = 100%	

Table 4.5: Participant 3 Experiment Results (Encase)

Dataset Group (2)	Probo Participant 3	
	Relevant (User)	Not Relevant (User)
Relevant (Correct: 39)	39	
Not Relevant (Correct: 33)		33
Total	39	33
Error Rate = (FP+FN)/ (Total Number of Items)	0	0
Error Rate Avg. This indicates the Total Errors' Avg.	0	
True Positive (TP): Indicates the number of items selected correctly as relevant.	39	33
True Negative (TN): Indicates the number of items selected correctly as relevant.	33	39
False Positive (FP): Indicates the items incorrectly selected as relevant.	0	0

False Negative (FN): Indicates the number of items incorrectly selected as non-relevant.	0	0
Precision: TP/(TP+FP)	$39/(39+0) = 1$	$33/(33+0) = 1$
Recall: TP/(TP+FN)	$39/(39+0) = 1$	$33/(33+0) = 1$
F-Measure: (2*Precision*Recall)/(Precision+Recall)	1	1
Avg. Accuracy = 1 - Avg. Error Rate	$1 - 0 = 100\%$	

Table 4.6: Participant 3 Experiment Results (Probo)

4.1.4. Participants Combined Result Averages

	Encase/dtSearch						Probo			
	Participant 1	Participant 2	Participant 3	Participant 1	Participant 2	Participant 3	Participant 1	Participant 2	Participant 3	
Relevant (User)	39	33	17	52	40	32	33	39	33	39
Not Relevant (User)	39	33	17	52	40	32	33	39	33	39
Total (Correct Count)	40	32	39	33	40	32	39	40	32	39
True Positive (TP)	38	31	17	33	40	32	32	39	32	39
True Negative(TN)	31	38	33	17	32	40	32	32	39	33
False Positive (FP)	1	2	0	22	0	0	1	0	1	0
False Negative (FN)	2	1	22	0	0	0	7	1	0	0
Precision	0.974	0.939	1	0.600	1	1	0.970	0.821	1	0.970
Recall	0.950	0.969	0.436	1.000	1	1	0.821	0.970	0.975	1
F-Measure	0.962	0.954	0.607	0.750	1	1	0.889	0.889	0.987	0.985
Error Rate	0.042	0.042	0.306	0.306	0	0	0.111	0.111	0.014	0.014
Error Rate Avg.	4%		31%		0%		11%		1%	
Accuracy Avg.	96%		69%		100%		89%		99%	
Overall Time Spent to Complete Task (Minutes)	55		93		59		28		42	
Avg. Time Spent Per Tool (Minutes)	69						33			

Table 4.7: Participants Combined Result Averages

4.1.5. Percentage of Change (Time Analysis)

According to Microsoft (2012):

$$\text{Percentage of Change (\%)} = ((\text{New Value} - \text{Old Value}) / \text{ABS}(\text{Old Value}))$$

		Keyword Search Questions (Minutes)	Relevancy Question (Minutes)	Total (Minutes)
Encase	Participant 1	17	38	55
	Participant 2	64	29	93
	Participant 3	21	38	59
	Total (Minutes)	102	105	207
	Tool Avg. Total	34	35	69
Probo	Participant 1	3	25	28
	Participant 2	6	36	42
	Participant 3	3	25	28
	Total (Minutes)	12	86	98
	Tool Avg. Total	4	29	33
Percentage of Change (%)^{*-1}	$((4-34)/\text{ABS}(34))^*-1 = 88\%$	$((29-35)/\text{ABS}(35))^*-1 = 18\%$	53%	

Table 4.8: Percentage of Change (Time Analysis)

4.2. Interview & Survey Results

In this section all the participants' answers for the interviews, surveys, and questionnaires were combined.

Interviewee Names:

Participant 1: Captain Abdalla Mohamed Al Ali (EEEC)

Participant 2: Captain Ghayda Ali Abdulla (EEEC)

Participant 3: Major Wafa Naser (EEEC)

4.2.1. Interview Combined Transcripts (Pre-Tool Testing)

Thank you for participating in this study, please answer the following questions.

1. Can you please state your name and job title?

Capt. Abdalla Al-Ali: My name is Abdalla Mohamed Al-Ali and I work as a Digital Forensic Examiner.

Capt. Ghayda Ali: My name is Ghayda Ali Abdullah and I'm the head of the Audio & Video Forensics Section at the Emirates Electronic Evidence Center in the CID of Abu Dhabi Police.

Major Wafa Naser: My name Wafa Naser and I work as a Digital Forensics Examiner at the Emirates Electronic Evidence Center in Abu Dhabi Police.

2. Could you share your education background?

Capt. Abdalla Al-Ali: I have a Bachelor degree of Electronic Engineer from Khalifa University also a Masters degree in Information Security (Cyber Crime) from Zayed University.

Capt. Ghayda Ali: I received my Masters degree from Zayed University in Cyber Security. I got my Bachelors also from Zayed University in Information Security with a concentration in Networking.

Major Wafa Naser: I have a Bachelor degree of Information Security and Networking from Zayed University, I also have a Master degree in Information Security (Cyber Security) from Zayed University.

3. Could you share your professional background? And how long you've worked at the Center?

Capt. Abdalla Al-Ali: I have been working in the Law Enforcement field for 8 years as a Digital Forensic Examiner, where I extract the evidence of most digital devices types in order to write the result of examination in reports and submit them to different Courts in the UAE.

Capt. Ghayda Ali: I joined the Center in 2009 after I finished my Bachelors and got training courses inside and outside the country with different Law Enforcement agencies, and different forensic software companies. I took introductory and advanced courses in forensics in different sections in Computer, Mobile, and Audio & Video. I worked now for almost 9 years.

Major Wafa Naser: I have been working in the Law Enforcement field for 9 years as a Digital Forensics Examiner, where I extract the evidence of most digital devices in order to write the result of examination in reports and submit them to the court to take action.

4. Can you explain who your main customers are and the nature of services you provide as a Center?

Capt. Ghayda Ali: We serve entities from all over the Emirates; we've also received cases from Dubai. We serve the General Headquarters (GHQ) of Abu Dhabi Police, with different Departments, and the Ministry of Justice. We also serve cases from the Social Support Center, all different police stations, and the different departments of the CID. As for the services, we examine different types of digital evidence that might relate to cases or that can be found in the crime scene.

5. Can you take me through the digital forensic examination process you follow when a case is received in the Center?

Capt. Ghayda Ali: When we receive any case in the Center first we have to check that an examination authorization letter is included that would give us the legal authority to examine devices in the case. The customer should fill the service request information pertaining to the type of service they require be done on the submitted devices.

After that the case devices will be labeled, sealed in special evidence bags and stored in the evidence storage until it's time for it to be processed. When an examiner is available to process the case, they will sign it out of the store and start by photographing and disassemble the device if necessary. Afterwards, two images (forensic copies) will be created of the device's storage, one image would be the Master copy which would be sealed and sent to storage along with the original device, and the other image would be the working copy which the examination will be conducted on.

Later the image (working copy) will be taken to the forensic workstation that was readied beforehand to process the image. The forensic workstation will be prepared by restoring it to a known clean state that has no previous case data and that has all the forensic software preinstalled. There are certain verification processes that are run on the forensic workstation that I will not get into details of; I think it's enough to mention that it involves verification of both hardware and software, which is done after each restore process.

The examiner will then start working on the image on the workstation and do their analysis according to the customer service request. Finally, after the case analysis is completed a case report will be created to provide the customer with information about the analysis done and recovered information from the device. However, before a report is submitted a peer review and auditing processes are run on the case to ensure the soundness of the findings and how it was reached.

6. Have you worked on cases where you had to review many text files?

Capt. Abdalla Al-Ali: Yes, most of the cases require reviewing text files.

Capt. Ghayda Ali: Yes, many cases.

Major Wafa Naser: Almost all the cases require reviewing text files.

7. On average, how many text files would occur in a case? And do you usually have to review all them?

Capt. Abdalla Al-Ali: Around thousands, usually I review the important text files depending on the case.

Capt. Ghayda Ali: It can reach up to thousands of files, and it would depend on the nature of the case. So sometimes I might have to review all of them or most of them, but I might also rely on doing Keyword Searches so when I get all of those text files I have keywords I'm interested in based on the case type.

Major Wafa Naser: It depends on the number of exhibits you have and how big they are in each case. For example, if the case has (500 GB HDD) it could include not less than hundreds of thousands of text within it. Reviewing them depends on the case type and usually I go through most of them or use specific keyword search and filters to help me keep focused on the related ones.

8. Can you share which forensic tools you prefer to use? And why?

Capt. Abdalla Al-Ali: It depends on the type of the case, but I am familiar with Encase, Blade and on some cases Nuix, I use those tools because it's the standard in our lab, they offer many useful features, and they're relied on by most of the other law enforcements lab.

Capt. Ghayda Ali: For the analysis I prefer to use Encase, because it provides me with the features I want to use to conduct my examination. I'll give you an example, when I'm doing file carving I know how the process is done and which type of file it's looking for, also when I'm doing file signature analysis I know how that type of analysis is done and the results I would expect to find.

Major Wafa Naser: Per case I use multiple tools, however the main forensic tool I prefer using is "Encase from guidance software". Because it's simple and has the options I need, I also use third party tools such as, "Blade", "Recuva", "IEF", and "Nuix".

9. How do you start your examination once the forensic image and workstation are ready?

Capt. Abdalla Al-Ali: First of all adding the image to the examination tool, verifying image, mounting image, indexing, search for keywords, verifying file signature, calculating hash values, carving (deleted items and lost files), finding related evidence and writing the report.

Capt. Ghayda Ali: After the forensic workstation is ready, and depending on the software I'm using and as I mentioned before I prefer to use Encase to do the analysis for the case. The work depends on case itself, for example if they are looking for a specific picture it means that I'll start by checking pictures, if they're looking for possible malware infection I'll do a virus scan and also check for Internet artifacts. But when sometimes I'm asked to do carving I prefer to use X-Ways because I like the way it exports the carved files.

Major Wafa Naser: I start with adding the image to the examination tool, verifying image, mounting image, indexing, search for keywords, verifying file signature, calculating hash value, carving (deleted items and lost files), finding related evidence and the writing report. Additionally, it depends on the type of the case; however, certain steps must be followed. Such as verification, recovering folders, file mounting, verifying file signatures and recovering deleted files. After that I start my analysis and review the results or do more processes depends on what evidence I found.

10. Do you find all case related information in the case request? If not, where else do you look for case related information?

Capt. Abdalla Al-Ali: Some cases are related to request, but mostly we need to ask the investigator or prosecutor to give us more details.

Capt. Ghayda Ali: Most of the time I can't find all case related information in the case request, so either I ask for the police case summary or I'd communicate directly with the customer, i.e. prosecutor or police investigator handling the case, and ask them about the WH questions to be able to work on it.

Major Wafa Naser: Some cases are related to request, but mostly we need to ask the investigator or prosecutor to give us more details. It depends on the request, I start analyzing either by reviewing the files in relation with the timeline, using certain keyword search and conduction further analysis through checking the computer logs and registry files. However, most of the cases we request the help of the investigator or prosecutor to give us more details.

11. What are the WH questions?

Capt. Ghayda Ali: Like the: what happened in the case? When did it happen? How it happened? Where? With such questions answered I'd be able to get more case details that I could use during my analysis.

12. Is there a feature to group some of your similar findings in your preferred forensic tools? If yes, explain it.

Capt. Abdalla Al-Ali: Yes, like the keyword search technique where if you have a list of words in one case you can use this list in different cases in order to find the files you are looking for. Another technique could be the hash value where in this method you calculate the hash value for the files like MD5 or SHA1 where there is no chance that two files have the same hash value, but the problem with this method is any change of data inside the file, even just adding one character, will give you a totally different hash value.

Capt. Ghayda Ali: No actually, and I hope that someday there will be something that could give us that feature.

Major Wafa Naser: We manually create our own keyword list per case and run it on the case through "Encase" which is the main forensics tool we use. Also, we can use another feature which is Hash library, where we run hashes for group of selected file on the whole case file.

13. Can you share your thoughts on a file grouping technique you think might be useful?

Capt. Abdalla Al-Ali: Maybe a Synonyms technique, where the examiner after he indexes the file and generates the keyword list on that file, he can use the Synonyms (words with the same meaning) in cases where the suspect changes the content of the file, also the user could use short words instead of long word. For example, if he wants to search for the word (programming) he can use a word like (program) to give himself more options like programmable, programmatic, program and programmer.

Capt. Ghayda Ali: We can do something like a Keyword Search and export the file and later I can use it for the same type of cases. For example, if the case is about terrorism I'd use certain words and export findings and use them for another case investigation. And also, I can do an MD5 hash for certain files, but while it can be useful it has a problem in that a search using MD5 hash would miss any files that were altered in anyway. This means that unless the files I'm searching for is the exact match to the MD5 they will be ignored by the software.

Major Wafa Naser: If I have technique which could understand the meaning of the keyword entered. In other words, to show me the results of not only the exact keyword but also any keywords that has the same meaning.

14. Is there a way to use information you've collected from a closed case in a new investigation? If yes, please explain it.

Capt. Abdalla Al-Ali: Yes, hash value, filenames, and keywords list.

Capt. Ghayda Ali: As I said, the Keyword Search can be used for future cases even if the current case is closed, and I can make a library that I can use for similar case types. Also, the MD5 hash can be used, but keep in mind what I said about the difficulty of the exact match problem.

Major Wafa Naser: Yes, Keyword list and hash values, however it's not always accurate since that hash value could be changed if the file was altered.

15. Can you share your thoughts on a way that information gathered from a closed case be used in a new investigation?

Capt. Abdalla Al-Ali: Build a database where the cases will be categorized depending on the types such as drugs case. For example, when a new drug case is added to the database it will match any similarity to old cases.

Capt. Ghayda Ali: Maybe I can use the Keyword Search. For example, if I have a drug case and I use certain words in the case to find some files or some emails, I can use the same Keyword Search for future cases. But if I create MD5 hashes for some of those files I will not be able to find something that had been altered with as little change as a word within the file. Because the MD5 is looking at the whole file that has the same MD5, so if there was any change on it the MD5 for that file will be different.

Major Wafa Naser: A tool that creates a list of categorized topics based on the case type. For example, names, contacts, usernames, locations, and other related information per case.

16. Do you have anything else you'd like to add?

Capt. Abdalla Al-Ali: No.

Capt. Ghayda Ali: Actually, I hope that if there were people interested in doing research on such problems, I would ask them to try and create software that can help us find the same text file's content or the same words that's been used in different text files because they would make our lives easier.

Major Wafa Naser: No.

Thank you for your participation.

4.2.2. Survey (Post-Tool Testing)

In this section, the survey answers from all the participants were combined.

Based on the tests you've done of the developed tool, when compared to your current preferred forensic tool. If you think of a case that contains a lot of text based files:

1. Which tool do you think might lead you to useful results faster during an investigation?

Capt. Abdalla Al-Ali: The developed tool.

Capt. Ghayda Ali: The tool I test is very useful

Major Wafa Naser: The tested tool (Probo).

2. Did you encounter similar grouping techniques in any other tool than the developed tool?

Capt. Abdalla Al-Ali: No, only with the developed tool.

Capt. Ghayda Ali: I tried Encase, but it did not have similar techniques. I also tried dtSearch, which could do some of the techniques (keyword search) but not all. The tested tool was much better and faster and easier to use.

Major Wafa Naser: No.

3. Do you think the developed tool and the techniques its promoting could be a good addition to have in any of the current forensic tools?

Capt. Abdalla Al-Ali: Yes, it will be a good addition.

Capt. Ghayda Ali: Sure.

Major Wafa Naser: Extremely useful and should be added, because it does automatic grouping.

4. Do you have any suggestions to improve the developed tool?

Capt. Abdalla Al-Ali: No.

Capt. Ghayda Ali: Maybe if it could do similar techniques for pictures, video, and audio files. For example, I can build a DB for some type of case and then run it on a new case and it would give me results faster.

Major Wafa Naser: If it could let the examiner create his own group by adding his keyword list (i.e. Drugs or terrorism) based on his experiences and preferences.

Thank you for your participation in this study. Do you have anything else you'd like to add?

Capt. Abdalla Al-Ali: No.

Capt. Ghayda Ali: No.

Major Wafa Naser: Amazing tool, very useful for Forensic examiners, especially the huge amount of data encountered these days. The examiner analysis could be spent on a smaller set of data and utilize the time spent on bigger data.

Thank you for your participation.

4.2.3. Questionnaire Results

The participants were asked to complete a list of close-ended questions for their opinion on the developed tool after they completed the experiments (Figures 4.1 – 4.3). The results indicated that the participants agreed on the quality of the developed tool with an overall rating of 5 out of 5, as shown in (Figures 4.4 – 4.6).

Questions

Interviewee Name: Abdullah Al-Ali

Post Tool Testing

Based on the tests you've done of the developed tool, when compared to your current preferred forensic tool. If you think of a case that contains a lot of text based files:

1. Which tool do you think might lead you to useful results faster during an investigation?

The developed tool

2. Did you encounter similar grouping techniques in any other tool than developed tool?

No only at developed tool

3. Do you think the developed tool and the techniques its promoting could be a good addition to have in any of the current forensic tools?

Yes it will be good addition

4. Do you have any suggestions to improve the developed tool?

No

Ending: thank you for your participation in this study. Do you have anything else you'd like to add?

Thank you.

Figure 4.1: Filled Up Survey (Participant 1)

Questions

Interviewee Name:

Ghazala Ali

Post Tool Testing

Based on the tests you've done of the developed tool, when compared to your current preferred forensic tool. If you think of a case that contains a lot of text based files:

1. Which tool do you think might lead you to useful results faster during an investigation? The tool I test is very useful

2. Did you encounter similar grouping techniques in any other tool than developed tool? I try Encase it did not has similar techniques, also I try dtsearch it can do some of techniques but not all and the tested tool was much better and faster and easier to use

3. Do you think the developed tool and the techniques its promoting could be a good addition to have in any of the current forensic tools?

Some

4. Do you have any suggestions to improve the developed tool?

if it can do similar techniques for pictures and audio files, for example, I can build DB for some type of case and then run it to compare, case it give me fact result

Ending: thank you for your participation in this study. Do you have anything else you'd like to add?

Thank you.

Figure 4.2: Filled Up Survey (Participant 2)

Questions

Interviewee Name:

Wafa

Post Tool Testing

Based on the tests you've done of the developed tool, when compared to your current preferred forensic tool. If you think of a case that contains a lot of text based files:

1. Which tool do you think might lead you to useful results faster during an investigation?

The tested tool (probo)

2. Did you encounter similar grouping techniques in any other tool than developed tool? No

3. Do you think the developed tool and the techniques its promoting could be a good addition to have in any of the current forensic tools?

Extremely useful and should be added, because it ~~leaves~~ does automatic grouping.

4. Do you have any suggestions to improve the developed tool?

If it could ~~teach~~ ~~that~~ ~~the~~ let the examiner create his own group by adding his keywords list (Dvst) or (terrorist)

Ending: thank you for your participation in this study. Do you have anything else you'd like to add?

Amazing tool, very useful for forensic examiner especially ^{based on his experience and preference} with the huge amount of data encounter these

Thank you.

days. The examiner analysis could be spent on smaller set of data and utilize the bigger time spent on bigger data.

Figure 4.3: Filled Up Survey (Participant 3)

EVALUATION CRITERIA OF PARTICIPANT'S VIEW

Participant Name: Abdullah Al-Ali
 Participant Signature: [Signature]

For each item identified below, circle the number to the right that best fits your judgment of its quality. Use the scale above to select the quality number.

Item	Scale				
	P o o r	F a i r	N e u t r a l	G o o d	E x c e l l e n t
1. How useful was the content to the investigation?	1	2	3	4	5
2. How appropriate was the clustering of information?	1	2	3	4	5
3. Did the suggested documents contribute to the overall solving of the investigation needs?	1	2	3	4	5
4. How relevant were the suggested documents to requested user query or topic of interest?	1	2	3	4	5
5. Discovery of new information that relate to the investigation	1	2	3	4	5
6. Ease of use	1	2	3	4	5

Figure 4.4: Filled Up Questionnaire (Participant 1)

EVALUATION CRITERIA OF PARTICIPANT'S VIEW

Participant Name: Ghazala Ali
 Participant Signature: [Signature]

For each item identified below, circle the number to the right that best fits your judgment of its quality. Use the scale above to select the quality number.

Item	Scale				
	P o o r	F a i r	N e u t r a l	G o o d	E x c e l l e n t
1. How useful was the content to the investigation?	1	2	3	4	5
2. How appropriate was the clustering of information?	1	2	3	4	5
3. Did the suggested documents contribute to the overall solving of the investigation needs?	1	2	3	4	5
4. How relevant were the suggested documents to requested user query or topic of interest?	1	2	3	4	5
5. Discovery of new information that relate to the investigation	1	2	3	4	5
6. Ease of use	1	2	3	4	5

Very Easy

Figure 4.5: Filled Up Questionnaire (Participant 2)

EVALUATION CRITERIA OF PARTICIPANT'S VIEW

Participant Name: *Wafa*

Participant Signature: *[Signature]*

For each item identified below, circle the number
to the right that best fits your judgment of its quality.
Use the scale above to select the quality number.

Item	Scale				
	P o o r	F a i r	N e u t r a l	G o o d	E x c e l l e n t
1. How useful was the content to the investigation?	1	2	3	4	5
2. How appropriate was the clustering of information?	1	2	3	4	5
3. Did the suggested documents contribute to the overall solving of the investigation needs?	1	2	3	4	5
4. How relevant were the suggested documents to requested user query or topic of interest?	1	2	3	4	5
5. Discovery of new information that relate to the investigation	1	2	3	4	5
6. Ease of use	1	2	3	4	5

Figure 4.6: Filled Up Questionnaire (Participant 3)

	Scale				
	P o o r	F a i r	N e u t r a l	G o o d	E x c e l l e n t
Participant 1: Captain Abdalla Al-Ali					
1. How useful was the content to the investigation?	1	2	3	4	5
2. How appropriate was the clustering of information?	1	2	3	4	5
3. Did the suggested documents contribute to the overall solving of the investigation needs?	1	2	3	4	5
4. How relevant were the suggested documents to requested user query or topic of interest?	1	2	3	4	5
5. Discovery of new information that relate to the investigation	1	2	3	4	5
6. Ease of use	1	2	3	4	5
Participant 2: Captain Ghayda Ali					
1. How useful was the content to the investigation?	1	2	3	4	5
2. How appropriate was the clustering of information?	1	2	3	4	5
3. Did the suggested documents contribute to the overall solving of the investigation needs?	1	2	3	4	5
4. How relevant were the suggested documents to requested user query or topic of interest?	1	2	3	4	5
5. Discovery of new information that relate to the investigation	1	2	3	4	5
6. Ease of use	1	2	3	4	5
Participant 3: Major Wafa Naser					
1. How useful was the content to the investigation?	1	2	3	4	5
2. How appropriate was the clustering of information?	1	2	3	4	5
3. Did the suggested documents contribute to the overall solving of the investigation needs?	1	2	3	4	5
4. How relevant were the suggested documents to requested user query or topic of interest?	1	2	3	4	5
5. Discovery of new information that relate to the investigation	1	2	3	4	5
6. Ease of use	1	2	3	4	5

Table 4.9: Combined Participants Questionnaire Results (Post-Experiments)