available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/diin

Digital Investigation

Digital forensic text string searching: Improving information retrieval effectiveness by thematically clustering search results

*Nicole Lang Beebe**, *Jan Guynes Clark*

The University of Texas at San Antonio, Department of IS&TM, One UTSA Circle, San Antonio, TX 78249, United States

Keywords:
Digital forensics
Text string search
Text clustering
Self-Organizing Map
Kohonen

ABSTRACT

Current digital forensic text string search tools use match and/or indexing algorithms to search digital evidence at the physical level to locate specific text strings. They are designed to achieve 100% query recall (i.e. find *all* instances of the text strings). Given the nature of the data set, this leads to an extremely high incidence of hits that are not relevant to investigative objectives. Although Internet search engines suffer similarly, they employ ranking algorithms to present the search results in a more effective and efficient manner from the user's perspective. Current digital forensic text string search tools fail to group and/or order search hits in a manner that *appreciably* improves the investigator's ability to get to the relevant hits first (or at least more quickly). This research proposes and empirically tests the feasibility and utility of post-retrieval clustering of digital forensic text string search results – specifically by using Kohonen Self-Organizing Maps, a self-organizing neural network approach.

This paper is presented as a work-in-progress. A working tool has been developed and experimentation has begun. Findings regarding the feasibility and utility of the proposed approach will be presented at DFRWS 2007, as well as suggestions for follow-on research.

1. Introduction

Textual evidence is important to the vast majority of digital investigations. This is because a great deal of stored digital data is linguistic in nature (e.g. human languages, programming languages, and system and application logging conventions). Some examples of important text-based evidence include: email, Internet browsing history (both logs and the content itself), instant messaging, word processing documents, spreadsheets, presentations, address books, calendar appointments, network activity logs, and system logs.

Digital forensic[1] text string searches are designed to search every byte of the digital evidence, at the physical level, to locate specific text strings of interest to the investigation. Given the nature of the data sets typically encountered, as well as the classic precision – recall trade-off problem,[2] text string search results are extremely noisy, which results in inordinately high levels of information retrieval (IR) overhead and information overload (Beebe and Dietrich, 2007). Frequently, investigators are left to wade through hundreds of thousands of search hits for even reasonably small queries (i.e. 10 search strings) on reasonably small devices (i.e. 80 GB) – most of

* *Corresponding author.*
E-mail addresses: nicole.beebe@utsa.edu (N.L. Beebe), jan.clark@utsa.edu (J.G. Clark).

[1] The terms "forensic" and "evidence" are used loosely in this paper. While such terms carry specific legal connotations, they are often used generically to refer to any type of digital investigation and resultant information derived from analytical efforts.

[2] The classic precision – recall trade-off problem suggests that achievable query precision decreases as query recall increases (Kowalski and Maybury, 2000).

1742-2876/$ – see front matter © 2007 DFRWS. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.diin.2007.06.005

which (i.e. 80–90% of the search hits) are irrelevant to investigative objectives. The investigator becomes inundated with data and wastes valuable investigative time scanning through noisy search results and reviewing irrelevant search hits.

There are fundamentally two classes of solutions to this problem: (1) decrease the number of irrelevant search hits returned, or (2) present the search hits in a manner which enables the investigator to find the relevant hits more quickly. The first solution class is disconcerting to many investigators and litigators, since it results in information reduction, presumably by some automated means. This can prohibit the investigator from finding important evidence.

The second solution class encompasses the basic approach that revolutionized web-based knowledge discovery: search hit ranking. This approach presents hits in priority order based on some determination of similarity and/or relevancy to the query. This approach is much more attractive to investigators and litigators, since it improves the ability to obtain important information, without sacrificing fidelity.

Current digital forensic text string search approaches fail to employ either solution class. They use simple matching and/or indexing algorithms that return all hits. They fail to successfully implement grouping and/or ranking algorithms in a manner that *appreciably* reduce IR overhead (time spent scanning/reviewing irrelevant search hits). Search hits are commonly grouped by search string and/or "file item[3]" and/or ordered by their physical location on the digital media. Such grouping and ordering are inadequate, as neither substantially helps investigators get to the relevant hits first (or at least more quickly). New, better approaches in the second solution class are needed.

This paper outlines current research-in-progress that aims to improve the IR effectiveness of digital forensic text string searches. The proposed approach presents search hits in a manner that helps investigators locate hits relevant to investigative objectives more quickly. Specifically, the on-going research tests the feasibility and utility of using self-organizing neural networks to thematically cluster text string search hits.

2. Literature review

A review of extant digital forensics literature disclosed no research into ways to improve the IR effectiveness of digital forensic text string searching, other than a generic call for the extension of text mining and information retrieval research to digital forensics (Beebe and Dietrich, 2007). The review did disclose a general consensus that industry standard digital forensics tools are *not* scalable to large data sets (i.e. gigabytes and terabytes), due to concerns about information overload, IR overhead and technological scalability (Roussev and Richard, 2004; Casey, 2004; Giordano and Maciag, 2002; Schwartz, 2000).

Certainly, the need to retrieve textual data from a large corpus of documents is not new. An entire field, known as *text mining*, has emerged in recent years from the larger field of information retrieval (IR). Text mining and IR fields enjoy over 30 years of research into locating and retrieving textual artifacts. The web-based information retrieval and knowledge discovery field also boast extensive research and technological advances in text retrieval.

The remainder of this section will discuss research in these fields, thereby laying the foundation for the text string search approach proposed in this paper.

2.1. Internet search engines

In exploring the extensibility of text-based IR from other fields, we will begin with Internet search engine research. During the 1990s, Google™ introduced a new Internet search engine that was able to prioritize search hits for the user. Google™ used five ranking variables: (1) PageRank (leverages web structure mining data), (2) query coincidence with anchor text, (3) proximity measures (mathematical similarity between query words and document words), (4) query term order (the assumption that the query terms are listed in descending order of importance), and (5) visual presentation characteristics (the assumption that visually conspicuous words in a file are more important than those that are inconspicuous) (Brin and Page, 1998).

The basic approach implemented by Google™ and the vast majority of Internet search engines permits the return of voluminous search hits, but orders them via relevancy ranking algorithms. Similar to Google's™ five-variable relevancy ranking algorithm, Internet search engines identify a small set of variables by which queries, documents, and their similarity can be characterized. The primary problem with extending this ranked list approach to digital forensic text string searching is the inextensibility of commonly used variables. Variables such as PageRank, hyperlink relationship, and anchor text data simply do not extend to typical digital forensic data sets, wherein only a portion of the data is web-based. To extend the ranked list approach to digital forensic text string searching, new ranking variables must be theorized and empirically validated.

2.2. Desktop search engines

Since the computer hard drive is such a prevalent evidence type in digital forensic investigations, it seems natural to explore file system (AKA desktop) search engine research and technology. File system search engines allow users to "browse" their computer just as they would "browse" the Internet for information. It is an emerging industry and field of research. Some current commercial desktop search engines include: Google Desktop, Yahoo! Desktop, Copernic Desktop, Spotlight, and X1. Academically developed and/or open-source desktop search engines include: Eureka (Bhagwat and Polyzotis, 2005), Connections (Soules and Ganger, 2005), and Semantic File System (Gifford et al., 1991).

File system search engines are functionally similar to Internet search engines in the sense that their basic process creates document indices and executes queries against inverted files. As with web-based IR, this approach makes query execution very fast.

[3] "File items" are chunks of data interpreted by the digital forensic software tool and presented to the user for review. They include allocated files, embedded objects, file system metadata files, blocks of unallocated and/or unpartitioned space, etc.

File system search engine technology is not extensible to digital forensic text string searching for two reasons. The first reason is the high startup costs of the index creation process. Initial indexing of one's file system can take days, and this is when only indexing a relatively small portion of the data on the physical device[4] (Boutin, 2004; Verton, 2004; Swartz, 2005). The second reason is that digital forensic text string searches seek to find digital evidence independent of the file system. They endeavor to find data at the physical level, as opposed to the logical, file system level. Therefore, any approach that relies on the file system is inextensible.

2.3. *Text mining research*

Text mining includes tasks designed to extract previously unknown information by analyzing large quantities of text, as well as tasks geared toward the retrieval of textual data from a large corpus of documents (Sebastiani, 2002; Fan et al., 2006; Sullivan, 2001). Several information processing tasks fall under the umbrella of text mining: information extraction, topic tracking, content summarization, information visualization, question answering, concept linkage, text categorization/classification, and text clustering (Fan et al., 2006). These are defined as follows:

- *Information extraction*: identifies conceptual relationships, using known syntactic patterns and rules within a language.
- *Topic tracking*: facilitates automated information filtering, wherein user interest profiles are defined and fine-tuned based on what documents users read.
- *Content summarization*: abstracts and condenses document content.
- *Information visualization*: represents textual data graphically (e.g. hierarchical concept maps, social networks, timeline representations).
- *Question answering*: automatically extracts key concepts from a submitted question, and subsequently extracts relevant text from its data store to answer the question(s).
- *Concept linkage*: identifies conceptual relationships between documents based on transitive relationships between words/concepts in the documents.
- *Text categorization/classification*: automatically and probabilistically assigns text documents into *predefined* thematic categories, using only the textual content (i.e. no metadata).
- *Text clustering*: automatically identifies thematic categories and then automatically assigns text documents to those categories, using only textual content (i.e. no metadata).

If applied to digital forensic text string searching, information extraction, content summarization, information visualization, and concept linkage would fit into the first solution class identified earlier (reduction of search results set size). These text mining approaches reduce the search result set size via data abstraction techniques, by and large.

[4] File system search engines index only logically resident (i.e. not permanently deleted) files. The data indexed include limited file metadata (e.g. filename, directory, and file type) and file content (although, often a set limit of the content).

The notion of topic tracking could be extended to digital forensic text string searching in the sense that a system could "learn" what hits are of interest to the investigator using relevancy feedback, and reprioritize remaining hits accordingly. This would fit into the second solution class identified earlier (present all hits, but grouped and/or ordered to decrease IR overhead). The problem, however, is that researchers have found that reliable filtering necessitates *extensive* relevancy feedback by the user (Yu and Lam, 1998).

Question answering does not fit well in either solution class, although the idea of a digital forensic tool automatically answering investigative questions is certainly intriguing!

Text categorization (AKA text classification) and text clustering fit well into the second solution class. Modern text categorization is the automatic, probabilistic assignment of text documents into a predefined set of thematic categories, using only the textual content (i.e. no metadata) (Sebastiani, 2002). Text categorization is a supervised, inductive learning process (i.e. a training set is required).

The problem with text categorization approaches in the digital forensics context is the requirement for supervision and the dependence on robust training sets. Each digital forensics case and each piece of digital evidence are unique. It is unrealistic to presume that investigators will have the time or the ability to assemble robust training sets needed for traditional inductive machine learning techniques.

Text clustering departs from text classification in that the algorithm automatically derives the thematic categories from the data. Text clustering is a form of unsupervised machine learning, and thus does not require training sets, extensive relevancy feedback, or supervision with respect to category identification – neither number nor theme. The unsupervised nature of text clustering and the resultant training independence makes it a promising solution to the digital forensic text string search problem.

Text clustering techniques are applied pre-retrieval and/or post-retrieval. The basic premise of pre-retrieval clustering is that document sets can be clustered thematically, and searches subsequently executed against document sub-sets that are thematically related to the query. Thus, pre-retrieval clustering fits into the first solution class – reduce search result set size.

The basic premise of post-retrieval clustering is that query precision (with respect to investigative objectives) can be improved by thematically grouping query results. Such grouping has been shown to help the user find relevant hits more efficiently. This is due to the *cluster hypothesis*: computationally similar documents tend to be relevant to the same query (van Rijsbergen, 1979).

Empirical research has repeatedly shown that clustered query results improve information retrieval effectiveness over traditional ranked lists using static relevance ranking models. Such results hold true for both traditional text-based information retrieval (i.e. clustering retrieved documents from a digital library) (Leuski and Allan, 2000, 2004; Hearst and Pedersen, 1996; Leouski and Croft, 1996; Leuski, 2001) and web-based information retrieval (Zeng et al., 2004; Zamir and Etzioni, 1998, 1999).

It is argued that thematic clustering of text string search hits will lead to separation between investigatively relevant

and investigatively irrelevant hits. For example, experienced investigators often refuse to search for the word ''kill'' in murder investigations, because of its high false-positive rate (investigatively speaking). The resource impact of such false positives is lessened greatly if all ''system kill'' and ''process kill'' hits could be automatically grouped together, and those constituent to human dialogue could be automatically grouped separately from the first group. The investigator could quickly bypass all of the false-positive ''kill'' hits by removing the ''system/process kill cluster'' from consideration after reviewing enough hits within the cluster to deem the entire cluster irrelevant to investigative objectives. This enables investigators to focus their attention on the ''kill'' hits that are more likely relevant to the investigation.

3. Proposed approach

Based on the literature review, the purpose of this on-going research is to test the feasibility and utility of thematically clustering digital forensic text string search results. To be feasible, a suitable clustering algorithm must be empirically shown to be extensible to digital evidence data sets – a fundamentally different data set and structure than those to which it was likely developed and historically applied. To establish extensibility, the algorithm must successfully cluster digital forensic text string search results via user-class computing platforms.

To ascertain the utility of clustering text string search results, the rate at which an investigator acquires like information from both clustered and un-clustered search results must be measured. If the clustered search results enable the investigator to find the relevant hits more quickly than when not clustered, then utility is demonstrated.

3.1. *Algorithm selection*

Data clustering methods can be grouped into five categories: partitioning, hierarchical, density-based, grid-based, and model-based (Han and Kamber, 2001). Each approach has advantages and disadvantages with respect to computational complexity, cluster quality, and ability to handle noisy data. Of the five categories mentioned, none of the partitioning, hierarchical, density-based, and grid-based methods are able to deliver high-quality clusters at low computational expense. And, only a few model-based clustering algorithms, when exposed to noisy data, are known to yield high-quality clusters at low computational expense.

The computational expense of model-based clustering approaches varies between approaches, but is often higher order with respect to input size, i.e. $O(n^2)$ (Roy et al., 1997). A notable exception is Kohonen's Self-Organizing Map (SOM) approach – an unsupervised neural network approach (Kohonen, 1981, 1989, 1990). SOM implementations usually scale linearly, $O(n)$, with data set size and sometimes even scale logarithmically, $O(\log(n))$ (Koikkalainen and Oja, 1990). They also handle noisy data well (Kohonen, 1990).

Researchers have empirically validated the applicability and use of SOMs to categorize the following types of textual documents: Internet homepages (Chen et al., 1996, 1998), document abstracts (Bote et al., 2002; Lin et al., 1999; Roussinov and Chen, 1998), electronic brainstorming/meeting discourse (Lin et al., 1999; Orwig et al., 1997), and newsgroup postings (Lagus et al., 1996).

The use of Kohonen SOMs for post-retrieval thematic clustering of digital forensic text string search results appears very promising. To improve scalability and performance, we specifically propose the use of Roussinov and Chen's (1998) Scalable Self-Organizing Map (SSOM) algorithm. The SSOM algorithm takes advantage of sparse matrix manipulation, thereby reducing computational expense and enabling the algorithm to scale to larger data sets.

4. Proposed methodology

The proposed text string search process has been instantiated in a working tool. As of this writing, the tool is complete[5] and experimental evaluation has begun. This section outlines the planned experimental methodology.

4.1. *Sample*

The proposed and developed text string search process/tool is being used to conduct a text string search over *real-world* digital evidence (in this case, a 40 GB hard drive from a civil suit previously investigated by a commercial digital forensics service provider).

The set of text search strings will be determined by an experiment volunteer. The volunteer is familiar with the case, given his employment with the commercial company providing access to the digital evidence. The volunteer is highly skilled and has over a decade of training and experience in conducting digital forensic investigations. Thus, the text string search list will be thorough, realistic, and appropriate to the case.

4.2. *Performance measures*

The same text string search will be conducted on the same evidence using three different tools: the tool/approach developed during this research and two industry standard digital forensics tools that string matching and indexing/Boolean-based algorithms (EnCase™ and FTK™, respectively). This will permit reporting of the IR effectiveness of the proposed approach relative to currently available approaches.

[5] The ''tool'' is actually a collection of scripts, open-source tools, modified open-source tools, and programs developed specifically for this project. The Sleuth Kit (TSK) (unmodified) and a modified version of Autopsy are being used to search for hits and extract logical files and unallocated clusters found to contain hits. Several C programs were written and are being called by a Perl program to extract ASCII strings from the extracted files and unallocated clusters, remove stop words, apply Porter's stemming algorithm, calculate term-document frequencies, and create document vectors. A C++ program was provided by Roussinov and Chen to conduct the SSOM clustering step. Several Perl scripts were written that ''glue'' all of these scripts and programs together. A final user interface was written using Ruby on Rails with a MySQL database housing the hits and associated metadata.

IR effectiveness in this context is a function of how much time must be spent scanning through and reviewing hits irrelevant to investigative objectives before getting to the relevant hits (more time spent reviewing irrelevant hits corresponds to poorer IR effectiveness).

If all three tools priority ranked their output (traditional rank lists), then the hit presentation order would drive IR effectiveness measures. In this case, however, we are testing the feasibility and utility of post-retrieval thematic clustering, as opposed to priority ranking search results. Therefore the order in which the hits are *reviewed* is the salient measure – not the order of presentation per se.

Ascertaining hit review order presents some significant methodological challenges. On the one hand, if one investigator reviewed the output of all three tools in their entirety, biases due to learning/practice effects and fatigue effects would result. On the other hand, significant error would be introduced if three different investigators reviewed the results from the three tools, due to individual differences in analytical approaches. Because of these issues, the order of hit review for EnCase(tm) and FTK(tm) will be presumed to be the order in which they are presented. It is important to note, however, that the presentation order will be aligned with text string priority as determined by the experiment volunteer. So, the first set of hits reviewed is presumed to be that corresponding to the investigator's highest priority text search string.

Similar assumptions cannot be made with the thematically clustered search hit results, since they will not be grouped by text string. Accordingly, a professional digital forensics practitioner (experiment volunteer) will evaluate the resultant search hits. His cluster and hit navigation order will be recorded electronically as he reviews the output.

The investigative relevancy of each and every hit in the superset of hits produced by all three tools will be determined by the researcher. This determination will be made based on context (AKA ''preview'') hit text (the hit and the ~60 bytes preceding and succeeding the hit). Methodological safeguards are planned to remove the possibility of bias associated with researcher relevancy determinations.

Armed with the investigative relevancy of all search hits and the hit review order for each tool's output, investigative recall and investigative precision will be measured at cut-off points (10% of hits reviewed, 20% of hits reviewed, etc.). The utility of the proposed approach will be a function of whether the investigator reviews (''gets to'') the investigatively relevant hits sooner than with EnCase™ or FTK™, and if so, to what degree. Specific measures are as follows:

$$\text{Query precision} = \frac{\#\text{ relevant hits retrieved}}{\#\text{ hits retrieved}}$$

$$\text{Query recall} = \frac{\#\text{ relevant hits retrieved}}{\text{Total }\#\text{ relevant hits in data set}}$$

A common search engine performance measure known as ''average precision'' (AvgP) will also be calculated. This measure evaluates an engine's ability to properly order relevant hits before irrelevant hits. The formula is as follows:

$$\text{AvgP} = \frac{\sum_{r=1}^{N} P(r) \times \text{rel}(r)}{R}$$

where r is the rank, N is the number hits retrieved, rel(r) is 0 or 1 (relevancy of hit), $P(r)$ is the total precision up to this point, R is the total number of relevant hits, and $P(r)$ is the total precision up to this point.

Finally, process time associated with each tool, as well as human analytic time will be measured. The proposed approach will certainly be more computationally expensive than current tools/approaches. The belief, however, is that by enabling investigators to find relevant information more quickly, an increase in ''computer time'' will be greatly eclipsed by the savings in ''human time'' spent analyzing search results.

5. Conclusion

This is a work-in-progress. Research is currently on-going. Anticipated challenges of the proposed research method are minimal, as several pilot experiments have been conducted during the development process with favorable results. The only unknowns at this point are whether the approach will indeed scale, and whether the resultant clustering will prove useful to the investigator. These findings and suggestions for follow-on research will be presented at the seventh annual Digital Forensic Research Workshop (DFRWS 2007).

Scalability is not a concern at this point, given the linearly scaled computational expense associated with the SSOM algorithm and the resource utilization and process times observed during pilot tests already conducted. Thus, feasibility is a low risk at this point.

The utility of thematically clustering digital forensic text string search results remains to be seen. However, initial pilot tests have shown extremely promising results with respect to cluster quality and the overall utility of the approach. Also, as previously stated, empirical research has demonstrated the superior performance of clustered search hits over ranked search hits. This research will determine whether this conclusion holds true in the digital forensic text string search context using real-world digital evidence.

REFERENCES

Beebe NL, Dietrich G. A new process model for text string searching. In: Shenoi S, Craiger P, editors. Research advances in digital forensics III. Norwell: Springer; 2007. p. 73–85.

Brin S, Page L. The anatomy of a large-scale hypertextual web search engine. Comput Netw. ISDN Syst 1998;30(1–7):107–17.

Bhagwat D, Polyzotis N. Searching a file system using inferred semantic links. In: Sixteenth ACM conference on hypertext and hypermedia, 2005, Salzburg, Austria.

Boutin P. Keeper finders: five new programs that let you search your hard drive without having a seizure. Slate; 2004.

Bote VPG, Anegon FdM, Solana VH. Document organization using Kohonen's algorithm. Inf Process Manag 2002;38:79–89.

Casey E. Network traffic as a source of evidence: tool strengths, weaknesses, and future needs. Digit Investig 2004;1:28–43.

Chen H, et al. Internet browsing and searching: user evaluations of category map and concept space techniques. J Am Soc Inf Sci 1998;49(7):582–603.

Chen H, Schuffels C, Orwig R. Internet categorization and search: a self-organizing approach. J Vis Commun Image Rep 1996; 7(1):88–102.
Fan W, et al. Tapping the power of text mining. Commun ACM 2006;49(9):77–82.
Giordano J, Maciag C. Cyber forensics: a military operations perspective. Int J Digit Evid 2002;1(2):1–13.
Gifford DK, et al. Semantic file systems. In: Thirteenth ACM symposium on operating systems principles SOSP '91, Pacific Grove, California.
Hearst MA, Pedersen JO. Reexamining the cluster hypothesis: scatter/gather on retrieval results. In: Nineteenth ACM international conference on research and development in information retrieval. Zurich, Switzerland: ACM Press; 1996.
Han J, Kamber M. Data mining: concepts and techniques. San Diego: Academic Press; 2001. p. 1–550.
Kowalski GJ, Maybury MT. Information storage and retrieval systems: theory and implementation. In: Croft WB, editor. The Kluwer intern international series on information retrieval. 2nd ed. Boston: Kluwer Academic Publishers; 2000. p. 1–318.
Kohonen T. Automatic formation of topological maps of patterns in a self-organizing system. In: Second scandinavian conference on image analysis, 1981, Espoo, Finland.
Kohonen T. Self-organization and associative memory. In: Huang TS, Kohonen T, Schroeder MR, editors. Springer series in information sciences. 3rd ed. Berlin, Heidelberg: Springer-Verlag; 1989. p. 1–312.
Kohonen T. The self-organizing map. Proc IEEE 1990;78(9):1464–80.
Koikkalainen P, Oja E. Self-organizing hierarchical feature maps. In: 1990 IJCNN international joint conference on neural networks, 1990, San Diego, California.
Leuski A, Allan J. Interactive information retrieval using clustering and spatial proximity. User Model User-Adap Interact 2004;14(2–3):259–88.
Leouski AV, Croft WB. An evaluation of techniques for clustering search results. Amherst, MA: Computer Science Department, University of Massachusetts at Amherst; 1996. p. 1–19.
Leuski A. Evaluating document clustering for interactive information retrieval. In: Tenth international conference on information and knowledge management. Atlanta, Georgia: ACM Press; 2001.
Leuski A, Allan J. Improving interactive retrieval by combining ranked lists and clustering in RIAO. College de France; 2000.
Lin C, Chen H, Nunamaker Jr JF. Verifying the proximity and size hypothesis for self-organizing maps. J Manag Inform Syst 1999;16(3):57–70.
Lagus K, et al. Self-organizing maps of document collections: a new approach to interactive exploration. In: Proceedings of the second international conference on knowledge discovery and data mining, 1996. p. 238–243.
Orwig R, Chen H, Nunamaker Jr JF. A graphical, self-organizing approach to classifying electronic meeting output. J Am Soc Inform Sci 1997;48(2):157–70.
Roussev V, Richard III GG. Breaking the performance wall: the cases for distributed digital forensics in Digital Forensics Research Workshop (DFRWS). Maryland, USA: Linthicum; 2004.
van Rijsbergen CJ. Information retrieval. 2nd ed. London: Butterworths; 1979.
Roy A, Govil S, Miranda R. A neural-network learning theory and a polynomial time RBF algorithm. IEEE Trans Neural Netw 1997;8(6):1301–13.
Roussinov D, Chen H. A scalable self-organizing map algorithm for textual classification: a neural network approach to thesaurus generation. Commun Cognit Artif Intell 1998; 15(1–2):81–111.
Schwartz M. Cybercops need better tools. Computerworld 2000;1.
Soules CAN, Ganger GR. Connections: using context to enhance file search. In: Twentieth ACM symposium on operating systems principles. Brighton, UK: ACM; 2005.
Swartz N. Google brings search to the desktop. Inform Manag J 2005;39(1):8.
Sebastiani F. Machine learning in automated text categorization. ACM Comput Surv 2002;34(1):1–47.
Sullivan D. Document warehousing and text mining – techniques for improving business operations, marketing, and sales. New York: John Wiley & Sons, Inc.; 2001. p. 542.
Verton D. Are desktop search programs ready for prime time? PC World 2004.
Yu KL, Lam W. A new on-line algorithm for adaptive text filtering. In: CIKM-98, seventh ACM international conference on information and knowledge management, 1998, Bethesda, MD.
Zeng H-J, et al. Learning to cluster web search results. In: Twenty-seventh ACM international conference on research and development in information retrieval. Sheffield, South Yorkshire, UK: ACM Press; 2004.
Zamir O, Etzioni O. Web document clustering: a feasibility demonstration. In: Nineteenth international ACM SIGIR conference on research and development of information retrieval. Melbourne, Australia: ACM Press; 1998.
Zamir O, Etzioni O. Grouper: a dynamic clustering interface to web search results. In: Eighth world wide web conference, 1999, Toronto, Canada.

Nicole Lang Beebe (nicole.beebe@utsa.edu) is a Doctoral Candidate and Instructor at the University of Texas at San Antonio, where she is working on her Ph.D. in Information Technology. She has nearly 10 years experience in information security and digital forensics. Her experience spans the commercial, government, and law enforcement sectors. She is a Certified Information Systems Security Professional (CISSP), a certified digital forensic examiner (EnCE), and holds degrees in electrical engineering (B.S.) and criminal justice (M.S.).

Jan Guynes Clark (jan.clark@utsa.edu) is a Professor at the University of Texas at San Antonio, which is a National Security Agency (NSA) designated Center of Academic Excellence. He is a Certified Information Systems Security Professional (CISSP), has a Ph.D. in Information Systems, and numerous publications on a variety of information systems topics.