Digital Forensic Text String Searching: Improving Information Retrieval Effectiveness by Thematically Clustering Search Results

By

Nicole Lang Beebe and Jan Clark

Presented At

The Digital Forensic Research Conference

DFRWS 2007 USA Pittsburgh, PA (Aug 13th - 15th)

Digital Forensic Text String Searching: Improving Information Retrieval Effectiveness by Thematically Clustering Search Results

DFRWS 2007

Department of Information Systems & Technology Management
The University of Texas at San Antonio

By Nicole L. Beebe &
Jan G. Clark, Ph.D.

August 13, 2007

Discussion Agenda

- Background
- Proposed Approach
- Experimental Methodology
- Data Analysis & Results
- Conclusion

Background

Digital Forensic Text String Search

- Searches evidence for text strings
  - Words, email addresses, numbers, etc.
- Current tools use literal search techniques
  - String matching algorithms
    - Search hits grouped by search string
    - Often ordered by file item and physical location
  - Full text indexing & Boolean queries
    - Search hits grouped by query
    - Often ordered by file item and physical location

Disadvantage

- Analytically burdensome
  - Hundreds of thousands of hits (or more)
  - 80%-90% of hits are not relevant
  - Result: High IR overhead
    - IR overhead is any time spent doing things other than reviewing relevant search hits
      - Query generation time
      - Search execution time
      - Time spent reviewing non-relevant search hits
  - Current grouping & ordering techniques do <u>not</u> *appreciably* lessen IR overhead

Research Question

Can IR & text mining algorithms be extended to digital forensic text string searching?

Extension Challenges

Traditional Contexts	DFTSS* Context
• Many, many searches	• Few searches
• Data sets grow incrementally	• Data sets are often unique to each case
• Logical level data only	• Logical & physical level
• Relatively homogeneous & structured data types	• Very heterogeneous & less structured data types
• High-end search engine platforms	• Relatively low-end search engine platforms

*DFTSS = Digital Forensic Text String Search

Research Question

To what extent does the extension of IR & text mining algorithms improve IR effectiveness of digital forensic text string searching?

“IR Effectiveness”

- IR effectiveness in DFTSS context
  - At or near 100% recall
  - Reasonable computational expense
  - Provides usual hit metadata
  - **Minimizes IR overhead**
    - **Keep search execution time reasonable**
    - **Minimize time spent reviewing non-relevant search hits**

Research Purpose

- Develop a better DFTSS process
  - Extend IR & text mining algorithms
- Evaluate IR effectiveness of new process
  - Build software tool
  - Compare against current processes
    - String match algorithm approach (EnCase™)
    - Indexing / Boolean-based approach (FTK™)

CAVEAT: This research is not a tool evaluation/comparison per se. It is meant to consider different/better ways to present text string search output. This approach could be "added on" to many digital forensic tools. The researcher is a happy consumer of both commercial tools listed!

Hypotheses

- New process outperforms current process WRT
  - Query precision rates
  - Query recall rates
  - Overall process time
    - Increased computer processing time eclipsed by savings in human analytical time

- Goal is to improve query precision & recall rates
  - Get to the investigatively relevant hits more quickly
  - A results presentation issue; not a fundamental change in the manner of the search

Proposed Approach

New Text String Search Process

Post-Retrieval Text Clustering

- Post-retrieval thematic clustering of search hits
  - – Unsupervised text mining approach
  - – Can be computationally efficient
  - – Improves IR effectiveness due to Cluster Hypothesis (van Rijsbergen 1979)
    - Computationally similar (clustered) documents tend to be relevant to the same query
    - Top performing cluster contains ≥ 50% of relevant hits (Hearst & Pedersen 1996)
  - – Outperforms traditional ranked lists
    - Hearst & Pedersen 1996; Leouski & Croft 1996; Leuski & Allan 2000; Leuski 2001; Leuski & Allen 2004

Text Clustering Algorithms

Qualities **Algorithms**	Low Computational Expense	Good Cluster Quality	Can Handle Noisy Data	Insensitivity to Input Order
Partitioning	X			
Hierarchical		X	X	X
Density-based		X	X	X
Grid-based	X		X	X
Model-based	VARIES	X	X	X

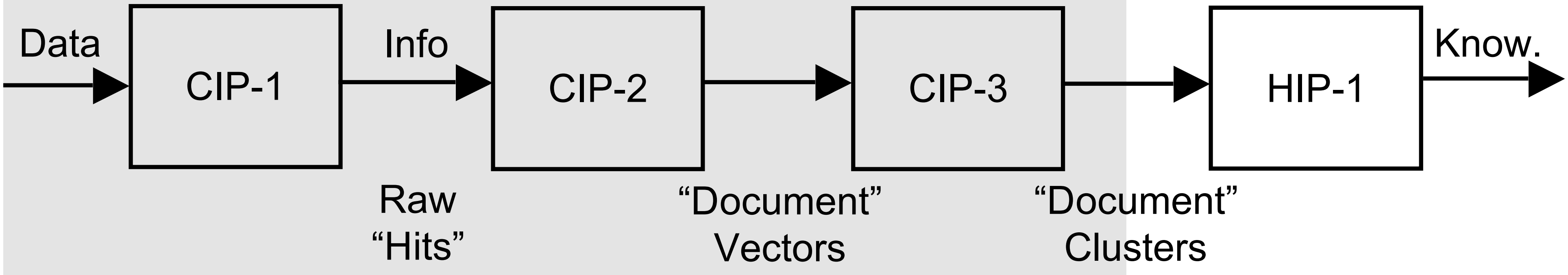
Self-Organizing NNets

- Qualities of self-organizing neural net (NNet)
  - Unsupervised machine learning method
  - Model-based clustering algorithm
  - Not computationally expensive
  - Demonstrated success in clustering data (text & non-text)
- Kohonen Self-Organizing Maps (1981)
  - Benefits
    - Low dimensional output (2-D)
    - Computationally efficient … *O(n)* to *O(log(n))*
    - Able to cluster textual & non-textual data

New Process

Computer Info Processing (CIP)

Human Info Processing (HIP)



Data

CIP-1

Info

CIP-2

CIP-3

HIP-1

Know.

Raw
"Hits"

"Document"
Vectors

"Document"
Clusters

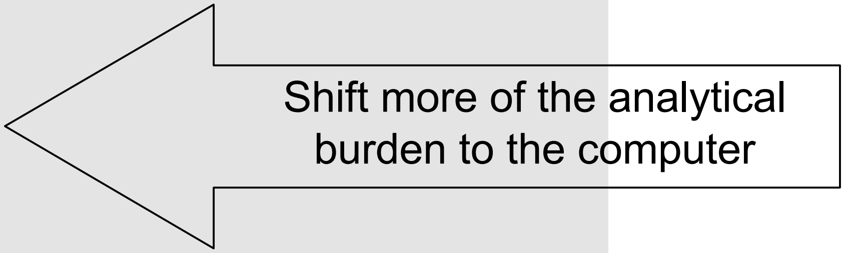
CIP-1:
String Search

CIP-2:
Prepare Data
for Clustering

CIP-3:
Cluster "Docs"
via Self-Organizing
Neural Network

HIP-1:
Manual Review
& Analysis of Hits

Increased CIP time should be eclipsed by HIP time savings.



Shift more of the analytical
burden to the computer

Experimental Methodology

Overview of Methodology

- Instantiate new process in prototype s/w tool
- Test hypotheses
  - Execute same search against same digital evidence using current & new processes
  - Measure IR effectiveness each time
  - Compare measures

Software Development

- S/W "Tool" developed was not an all-in-one tool
  - Series of s/w tools, scripts, & data handling procedures
  - Name for interface and general process: "Grouper"
- CIP-1: String search
  - Functionality
    - Locate all instances of text strings
  - Software development
    - Modified open source digital forensics tools
      - The Sleuth Kit (TSK) (C)
      - Autopsy (TSK's web-based interface) (Perl)

Software Development (cont.)

- CIP-2: Data preparation for clustering
  - Functionality
    - Identify “document” vocabulary
      - Extract all alphanumeric strings
    - Select a reduced dimension vocabulary
      - Apply stop word list (McCallum, *Bow* library)
      - Apply stemming algorithm (Porter, 1980)
    - Produce “document” vectors
  - Software development
    - Series of home-grown programs & scripts (C, Perl)
    - Porter’s open source stemmer (C)

Software Development (cont.)

- CIP-3: Clustering
  - Functionality
    - Thematically cluster “documents”
  - Software development
    - Selected Scalable SOM algorithm (Roussinov & Chen 1998)
      - Uses binary document vectors & sparse matrix manipulation
      - Much more computationally efficient that traditional SOMs
    - Code (C++) provided by Dr. Dmitri Roussinov, Ariz. State
      - Minor modifications made (debugging & output reformulation)

Software Development (cont.)

- HIP-1: Search result analysis
  - Functionality
    - Facilitate review of clustered search hits
      - Thematically clustered “documents”
      - Presents “documents” in priority order (similarity to cluster)
      - Presents search hits in order of physical location
    - Record key variables for IR effectiveness measures
      - Relevancy determinations, search hit review order, date/time stamps of user activity
  - Software development
    - Access database w/ Access programming & VB code

Hypotheses Testing

- Basic approach
  - Execute same search against same digital evidence using current & new processes
  - Measure IR effectiveness each time
  - Compare measures
- Test data sets
  - Real-world case (private forensics company)
    - Divorce case; allegations of extramarital activity; 40GB HD
  - Mock case (created by graduate students)
    - Murder case; allegation that wife caused husband's heart attack; 10GB HD (previously used & not wiped)

Hypotheses Testing (cont.)

- IR effectiveness
  - Measures
    - Query precision (accuracy)
    - Query recall (completeness)
    - Average precision (search engine performance score)
    - Time (computer & human info processing time)
  - Measurement points
    - Incremental cut-off points
      - 10% increments of # hits reviewed
    - Satisficing point (Simon 1947)
      - When elements of proof are satisfied, and/or
      - When all key textual artifacts have been located

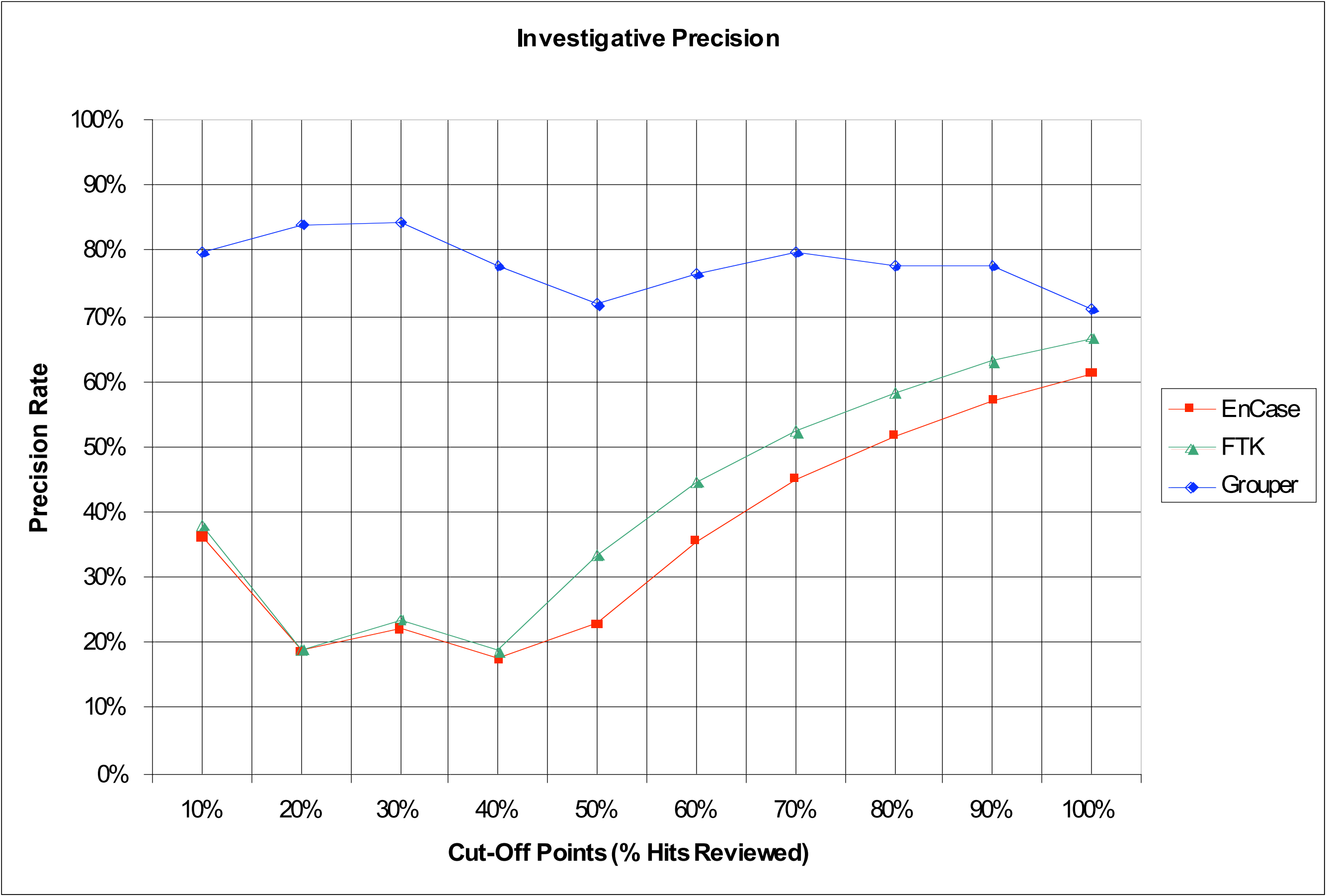
Data Analysis & Results

Real-World Divorce Case

Mock Murder Case

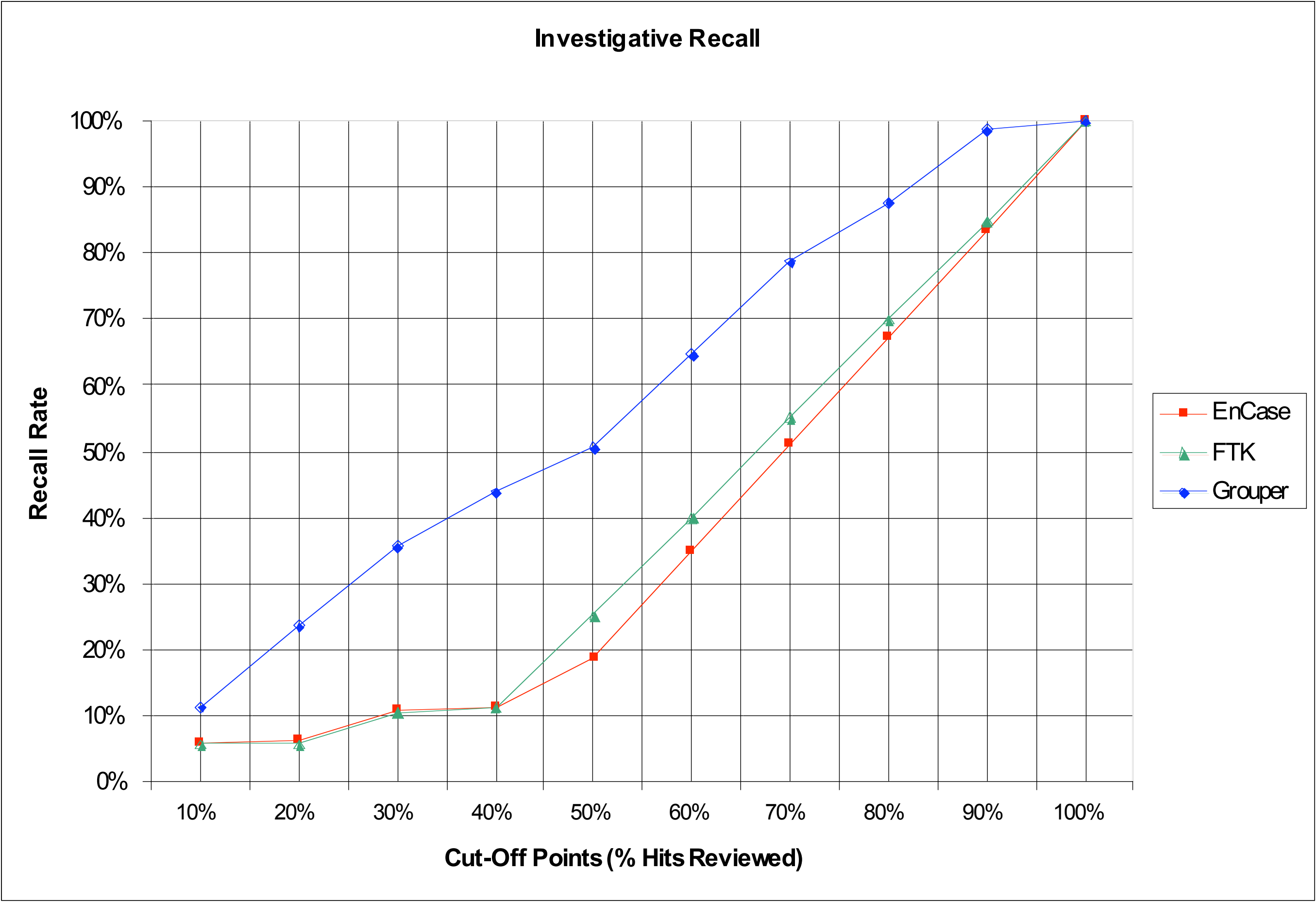
Real-World Case Results

- 17 search strings yielded ~25,000 search hits
- Query precision at incremental cut-off points



Real-World Case Results (cont.)

- Query recall at incremental cut-off points



Real-World Case Results (cont.)

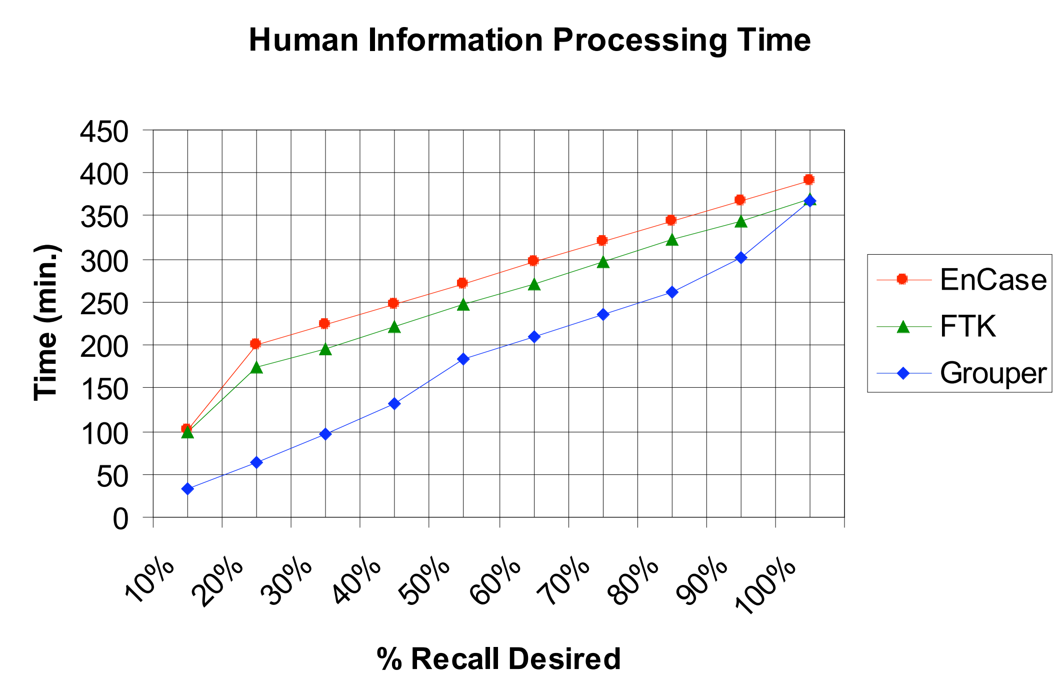
- Average precision scores
  - Score=0 : All non-relevant hits presented first
  - Score=1 : All relevant hits presented first

Tool	AvgP Score
EnCase™	0.432
FTK™	0.483
Grouper	0.781

- Conclusion
  - Post-retrieval clustering of search hits improves query precision & recall rate curves

Real-World Case Results (cont.)

- Process time
  - Additional computer info processing time: <20 min.
    - CIP-2 = 18.1 minutes (data preparation for clustering)
    - CIP-3 = 8 seconds (clustering step)
  - Savings in human analytical time observed



Conclusion:

Post-retrieval clustering of search hits improves overall process time

Real-World Case Results (cont.)

- Satisficing analysis
  - Motivation
    - Theory of administrative behavior (Simon 1947)
    - Investigators seldom review all search hits
      - Due to resource constraints and fatigue effects
  - Satisficing point determination
    - Clustered output
      - Subjectively determined by research volunteer
    - Current processes
      - Objectively determined by locating same digital artifacts in EnCase™ & FTK™ output as recovered up to satisficing point in clustered output

Real-World Case Results (cont.)

- Satisficing analysis results:

	EnCase™	FTK™	Grouper
Satisficing Cut-Off Point	99.15%	99.08%	**21.99%**
Precision	60.9%	66.3%	85.0%
Recall	98.3%	98.5%	26.4%
HIP-1 Time	388.4 min. (6.47 hrs)	366.8 min. (6.11 hrs)	62.1 min. (1.04 hrs)

Note: Satisficing points will vary between cases.

Mock Murder Case Results

- 19 search strings yielded ~110,000 search hits
- EnCase™ & FTK™ outperformed Grouper
  - WRT query precision & recall at most cut-off points
  - AvgP scores:
    - EnCase™ & FTK™ ~ 0.35
    - Grouper ~ 0.09
- Cluster quality statistics provide explanation
  - Cluster #48 (7x7 map)
    - Huge cluster containing vast majority of relevant hits
    - Very heterogeneous cluster content
    - Non-relevant hits generally presented first in this cluster
  - Map size is too small for search hit result set !!

Mock Murder Case Results (cont.)

- Tested explanation for poor IR effectiveness (SOM size too small)
  - Re-clustered “documents” into 20x10 map
  - Saw improved cluster quality statistics
    - Cluster #48 from 7x7 map split into 6 clusters (#181 & others)
    - Cluster #181 in 20x10 map still largest cluster, but
      - Noise level reduced
      - Improved cluster content homogeneity
      - Relevant hits now presented earlier

Murder Case Results (cont.)

- Simulated satisficing analysis results

	EnCase™	FTK™	Grouper (Map Size: 7x7)	Grouper (Map Size: 20x10)
Satisficing Cut-Off Point	58.5%	58.4%	75.9%	**32.6%**
Precision	15.0%	15.7%	10.2%	31.2%
Recall	93.7%	94.2%	65.3%	85.4%
HIP-1 Time	1,457.4 min. (24.29 hrs)	1,394.2 min. (23.24 hrs)	1,544.46 min. (25.74 hrs)	663.4 min. (11.06 hrs)

Conclusion

Study Conclusions

- Extension of scalable SOMs is feasible
  - Works on unique nature of DFTSS results
- Clustered search hits can improve IR effectiveness relative to precision & recall
  - Empirical results from real-world case support claim
    - >80% decrease in human analytical time
    - <20 minutes additional computer processing time
  - Empirical results from mock case do not, BUT
    - Predominantly because of insufficient SOM granularity
    - Simulated satisficing analysis results suggest larger map would support hypotheses regarding precision & recall

Study Conclusions (cont.)

- Clustering search hits can improve IR effectiveness relative to overall process time
  - Not cost prohibitive relative to clustering computer information processing time (CIP-2 & CIP-3)
  - Clustering can save human info processing time (more time than increased CIP-2 & CIP-3 time)

Limitations

- Generalizability
  - Only two cases studied
  - Problems with real-world data set precluded complete search; small search result set
  - Both hard drives were somewhat small
  - Single analysis/evaluator per case
- Reliance on open-source software
  - Necessary s/w design severely biased CIP-1 time and affected overall process time hypothesis testing
  - Not an “all-in-one” tool, as most are today

Contributions

- First academic work in improving IR effectiveness of DFTSS
- Theoretical extension of text mining research
  - Context varied
  - Extensibility wasn't guaranteed due to data set
- Practical implications
  - Makes an important analytical approach useful
  - Can reduce the incidence of missed evidence
  - Lessens impact of organizational resource constraints

Future Research

- Empirically validate larger map size findings
- Replication needed
- Studies needed re: SOM parameter optimization
- Consideration of parameter-less SOMs
- Research into more appropriate stop-word lists
- Cluster navigation behavior studies needed
- Studies to better understand satisficing points in digital forensics
- Better tool to further test time hypotheses

Comments or Questions?

Thank You!

nicole.beebe@utsa.edu