Electronic Discovery Reference Model

EDRM Search Guide

May 7, 2009

v. 1.17

Version	Date	Changed By	Changes
1.0	7/05/08	Venkat Rangan	Initial proposal submitted to EDRM Search Documentation Group, and checked into edrm.net
1.1	11/26/08	Jason Robman	Comments on sections 1, 2, 3, 8, 9
1.2	12/1/08	Stacy Monarko, Max Shoka	Section 3
1.3	1/5/09	Stacy Monarko, William Uppington, Linda Beaudoin, John Horan, Jay Di Silvestri, Terry Ahearn	Extended use case in Section 6
1.4	1/6/09	Venkat Rangan, Linda Beaudoin	Section 12
1.5	1/7/09	Jason Robman	Comments on Sections 7, 8
1.6	1/8/09	Venkat Rangan, Greg Buckles, Linda Beaudoin	Section 11
1.7	1/8/09	Andrew Szwez, Brian Weiss, Karen Williams	Sections 9 and 10 combined. Moved Meta Data section to Appendix. New text for Sections 9.1, 9.1.1, 9.2.1, 9.3.1. Removed XML examples.
1.8	1/9/09	Stacy Monarko, William Uppington, Linda Beaudoin, John Horan, Jay Di Silvestri, Terry Ahearn	Replaced Sections 6 and 7 with Extended Use Case
1.9	1/10/09	Venkat Rangan Greg Buckles	Incorporated Section 11 (previous Section 12) comments and moved material from new Section 11 to Appendix. Expanded Section 8.6 on Concept Search. Removed Section 5.
1.10	1/11/09	Terry Ahearn	Comments on Sections 1-3
1.11	1/11/09	Karen Williams	Added comments and general cleanup.
1.12	1/20/09	Adam Reilly	SQL Examples , comments on Language and Content section
1.13	1/20/09	Penny Apple	General copyediting
1.14	1/20/09	Chris Paskach, Brent Kidwell	Provided section on Search Framework
1.15	3/23/09	Ralph Losey	Comments on Appendix II.
1.16	5/7/09	John Wang	Added information on Index Term Selection, CJK Indexing, and RDBMS Indexing
1.17	5/7/09	Venkat Rangan	Added References section

1. Overview

During the discovery phase of litigation, parties to a dispute must take appropriate, reasonable steps to locate relevant, responsive Electronically Stored Information ("ESI") in response to discovery requests. This EDRM Search Guide focuses on the search, retrieval and production of ESI within the larger electronic discovery process described in the EDRM Model. Use of automated search can be an essential component in the e-discovery process as attorneys may perform automated searches to locate relevant, responsive, and/or privileged ESI for a legal matter. The commentary in this EDRM Search Guide describes the concept of search, the various search methodologies available to a practitioner in a particular case, and how various search methods may be deployed to achieve the most optimal results.

Typically, various automated searches are used by a Producing Party, in response to discovery request for ESI from a Requesting Party. The goal of this document is to provide a standardized way for the Requesting Party and Producing Party to communicate with each other regarding aspects of search that are important for a successful search, retrieval and production of responsive ESI. This document also addresses standardized reporting by which a Producing Party can provide search results to the Requesting Party. A production request may involve iterative evolution and refinement of searches between the Requesting Party and the Producing Party during the discovery lifecycle of the litigation. The EDRM Search Guide provides a mechanism for documenting this historical evolution of searches, thereby providing a vehicle for communicating these changes. Also, this document provides Best Practices Guidelines on when certain search methods are likely to produce more effective results, and aspects of search that may be considered to achieve optimal results. The EDRM Search Guide is properly viewed as a dynamic, living document that will require appropriate revision from time to time as the needs of the legal community develop and change and as the tools and techniques available to parties in disputes evolve to meet those needs.

2. Goals of this document

The overall goal of this EDRM Search Guide is to provide the legal community and e-discovery professionals with educational commentary and guidelines in search and retrieval methods. This document provides guidelines and suggestions to be considered by legal practitioners in developing appropriate and effective search methods. Use of automated search can be a critical component to the e-discovery process; however, utilizing automated search to locate responsive ESI to satisfy a Requesting Party's production request can also be complicated.

A further goal of this EDRM Search Guide is to offer a potential approach for requesting parties and producing parties to communicate with each other regarding the various aspects of the search methodology to be employed such as the search request parameters, tools employed to conduct the search, the results of the search and technical aspects of search. This EDRM Search Guide offers suggestions on potential ways in which a producing party can document and provide search results to a requesting party, including in iterative search processes, where such processes and communications are appropriate. A standard method for communicating these elements may facilitate a more effective dialogue among multiple parties involved in an e-discovery effort.

This EDRM Search Guide does the following:

a) Describes an overall search framework for e-discovery within the existing EDRM model;
b) Describes various search methodologies available to parties;
c) Describes how an EDRM search specification can be used as a protocol for communication between a requesting party and a producing party;
d) Specifies how a search dialog and historical negotiation information can be recorded;
e) Provides a vendor-neutral method of documenting and communicating searches;
f) Accommodates and provides for variations in levels of search implementations across vendors; and
g) Describes special technical issues the practitioner may wish to consider in designing the search process.

This EDRM Search Guide is not an attempt to substitute a formal technical system for the judgment, legal skills and expertise of attorneys involved in electronic discovery in any particular matter. Nor is it not the goal of this EDRM Search Guide to mandate a required or specific search methodology for addressing a particular discovery requests. Each case presents a unique set of facts and legal issues along with unique data sets. What may be an appropriate methodology in one case may be wholly inappropriate in another. This EDRM Search Guide does not describe which searches are, or are not, minimally effective or what searches are or should be acceptable by the courts.

It is also not a goal of this EDRM Search Guide to advocate or facilitate system-to-system transfer of search requests and results. Further, this EDRM Search Guide presents search specifications in a way that may or may not match any specific vendor's implementation. Parties who wish to use this EDRM Search Guide should be mindful that they may need to translate or map search specifications to the specific technical tools and methods available to them for executing the searches.

This EDRM Search Guide addresses some common issues encountered during e-discovery:

a) Specification of accurate search methodologies that a party intends to employ to respond to discovery requests;
b) Considerations in evaluating preliminary search results and/or data sampling to enable the parties (jointly or otherwise) to refine searches to address, among other things, presence of overly broad or overly restrictive search terms, search terms

with a high level of false positives, and other potential forms of the search terms that could be used to satisfy the discovery request;

c) In a collaborative process, the exchange of information regarding the processes each party has employed to perform searches in a manner so that each party has information about the technology, search specifications and any validation of the search process with an appropriate level of transparency;

d) Clear and accurate documentation of the search methodologies, including any iterative processes, and the final search parameters and results; and

e) Commentary and guidance for legal professionals to consider in designing and performing accurate and effective searches; and, to help make their search processes defensible.

3. Audience for this document

This EDRM Search Guide is intended to assist legal professionals who are involved in all phases of e-discovery during the litigation lifecycle.

3.1 Attorneys and Judges

This EDRM Search Guide will serve as a high-level reference point for attorneys and judges seeking to understand how and where search fits into the litigation lifecycle and to develop a background and understanding of search concepts. This EDRM Search Guide will provide specific terminology used in the area of search.

Several sections of this EDRM Search Guide, including Section 5, are particularly relevant for attorneys and judges. Section 5 explains how search fits into e-discovery. Section 6 provides useful commentary regarding common search methodologies used during e-discovery. In addition, Section 7 discusses issues that the legal professional may wish to consider in developing a more successful search protocol, such as technical information and concepts that may help to explain why some searches fail to yield relevant and responsive data, while other searches yield results that appear not to be responsive or relevant.

3.2 Corporate In-House Counsel

Given the trend for in-house counsel to take a more active role in e-discovery, this EDRM Search Guide will also serve as a valuable resource for corporate and litigation in-house legal counsel to develop an understanding of the process, opportunities, and challenges of e-discovery. Sections 4 and 5 of this EDRM Search Guide describes the search process. Section 6 of this EDRM Search Guide describes examples of how search may be implemented during litigation. The remaining sections of this EDRM Guide provide commentary on potential techniques and suggestions for effective search and search result validation.

3.3 Litigation Support Professionals and Paralegals

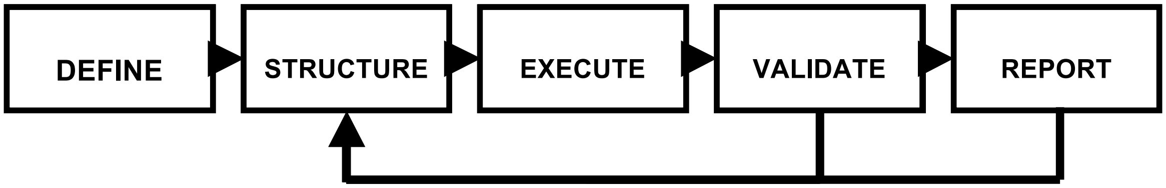
This EDRM Search Guide will be a resource for litigation support professionals and paralegals that are primarily responsible for actually performing the e-discovery tasks. The search framework and process outlined in Sections 4 and 5 of this EDRM Search Guide provide a structure for approaching these e-discovery tasks. Sections 6 through 8 of this EDRM Search Guide describe the various, techniques that can be applied when searching ESI. Finally, Section 9 of this EDRM Search Guide offers some leading approaches for testing and demonstrating the validity of the search process.

3.4 Litigation Service and Software Providers

This EDRM Search Guide will be a valuable resource for litigation support and software providers. Section 5 of this EDRM Search Guide provides a detailed overview of how search fits into the different stages of e-discovery. In addition, Sections 5 and 6 of this EDRM Search Guide offers a hypothetical use-case scenario to illustrate how search may be implemented during litigation from the perspective of the attorneys responsible for producing responsive information. Sections 7 and 8 of this EDRM Search Guide explain a variety of different search methodologies and techniques. Finally, Section 9 of this EDRM Search Guide illustrates how search results may be documented and preserved.

4. Search Framework

Key to understanding and explaining the search process, and therefore enhancing the transparency and reliability of that process, is a thorough understanding of the overall search process. Below is a high-level abstraction of the search process laying out the five major phases for performing an effective search in the context of E-Discovery. Each of the phases is described briefly below and in greater detail in the following sections of this paper.



The following sections elaborate this.

4.1 Define the search

The first step in the search process is to identify the purpose of the search exercise, and the goals to be achieved. A variety of reasons for conducting a search of data exist, ranging from a massive data reduction project to limit the scope of the document review, to pin-point identification of particular documents for purpose of an investigation or early-case assessment. Other goals of the search may be, for example, to identify potentially privileged documents, or potentially responsive (or unresponsive) material. Regardless of the specific purpose of the search, litigants can best maximize the success of the search process by defining the purpose and goals of the exercise. Considerations during this definition process include, but are not limited to:

- Identification of the target data that is being sought within the overall data corpus
- Identification of the desired results at the end of the search process (i.e., the defined deliverables)
- Exploration of the project's tolerance for risk of being under-inclusive or over-inclusive in search results
- Identification of cost constraints and budgetary considerations for the project
- Consideration of how the search process "fits" into the overall discovery, document production and case strategy
- Consideration of who is best positioned to conduct and implement the search - e.g., a vendor, outside counsel, in-house counsel and staff
- Understanding of how the case team will justify the completeness and reasonableness of the searches, and identification of the criteria that will be used to evaluate and substantiate the efficacy of the process

The process of defining the search is explored more in Sections 6 and 7 of this document.

4.2 Plan and Structure the Search

The next step in the search process is to plan and structure the search. Planning includes identification of the scope of data to be searched, who will conduct the search, and the technology and process that will be used to implement the search. For example, consideration should be given to the type of search which is most appropriate in the circumstances (keyword, concept, etc.), the use of text and/or metadata searches (e.g., searching for occurrence of terms in the text or fielded metadata like date ranges), and the output of the results of the search for testing and validation. Each of these topics are discussed in detail below in Section 6 (scope of the search), Section 8 (search methods) and Section 9 (search technology).

4.3 Execute the Search

Next in the search process is the execution of the search. All the definitional and planning work that preceded this step are now put to the test as the actual search is run using the technology tools on the assembled data. During execution the search is monitored, and the process and results of the search are captured, and documented. As a part of this process the results of the search may be communicated to the client or consumer of those results via a “hit” report, sample data, or other similar methods. The process of documenting the search process and results is discussed in more detail below in Section 10.

4.4 Validate the Search

Validation of the efficacy of the search process is paramount to insuring its comprehensiveness and effectiveness. At bottom, the validation steps seeks to determine if the search “worked” -- that is, did the search include all of the records that were to be searched and did it achieve the goals established during the definition phase described above. Details regarding this measurement and validation process are explored more fully in Section 11 below.

4.5 Report

Section 10 of this document explores in detail the manner in which the results of the search may be documented and presented. This documentation and reporting step is an integral aspect of presenting the decision-makers (e.g., the case team attorneys) with sufficient information to properly evaluate the results of the search, and the efficacy of the current search process. Absent clear and sufficient reporting, the attorneys and client will be unable to assess whether the search was sufficient to meet their case needs. And while the initial reporting should be considered iterative -- meaning the report may cause the search to be modified and re-run -- the final aspect of the search and reporting process will likely be documentary findings of the population of records that were searched, the search method and parameters used to execute the search and the results which could be shared with opposing counsel or the court as necessary to justify and explain the process.

5. Searches during EDRM Workflow Steps

Each step in the EDRM workflow requires deploying certain search techniques to speed up and automate the process. This section illustrates, with an example, how search is incorporated into the EDRM workflow.

5.1 Brief Background

This scenario involves two companies, Alpha Corporation and Beta Corporation, fierce competitors of one another in the field of Web-based record-keeping and analysis of insurance claims for medical treatments. Alpha has developed a technological advantage over Beta with its new Web 2.0 MedicEye monitoring technology – a combination of hardware, software, data storage and various analytical methodologies.

Alan Baker, a software developer and former employee of Alpha, has recently moved to Beta. Alan worked on a portion of the software development for Alpha's MedicEye. Beta has refused to comment on the hire, but Alpha suspects that Alan was hired to work on Beta's development of a competing offering. Alpha also suspects that Alan copied and took with him to Beta large amounts of source code, planning documents, financial analyses and market studies. Alpha further suspects that Beta has directed certain unnamed persons to hack into Alpha's computer system and obtain copies of some of the same sorts of materials Alpha suspects Alan has taken.

Alpha is based in Palo Alto, CA, and has offices in Chicago, Houston and Tokyo. Beta is based in Boston, MA and has offices in New York City, Paris and Abu Dhabi. For each company, most of the relevant documents and communications are in English, but some potentially important hard-copy materials and electronically stored information ("ESI") are in the respective local languages of each of the various non-U.S. offices.

5.2 Proux LLP Receives Miscellaneous ESI Collected by Isis LLP During Its Internal Investigation for Alpha's Board of Directors

(*i.e.*, "early" early case assessment)

Proux LLP is one of Alpha's regular outside counsel. On June 1, 2008, Paula Proux, a partner with Proux LLP, receives a call from Alex Arnold, General Counsel of Alpha. Alex tells Paula a little about Alpha's situation, including Alan's move from Alpha to Beta and Alpha's suspicion that Alan, and perhaps others at Beta's direction, have improperly taken copies of Alpha proprietary materials.

Alex also informs Paula that over the past several months, Ivan Isis from Isis LLP has been conducting an internal investigation for the Audit Committee of the Alpha Board of Directors, has interviewed a number of people, and has "informally" collected some hard-copy documents and ESI from a range of sources. Alex asks Paula to assemble a team, review the materials already collected, identify and interview potential witnesses, analyze the situation, and advise Alex how Alpha should proceed.[1]

Over the next several weeks, Ivan Isis from Isis LLP provides Paula with the following fruits of his internal investigation:

- A binder of short reports on interviews of key Alpha personnel, including a small number of relevant emails and related documents (in hard-copy form).

[1] As we'll see, Alpha ultimately sues Beta in the U.S. District Court for the District of Maryland – chosen because of the illustrative value of its "Suggested Protocol for Discovery of Electronically Stored Information" – and this suit proceeds far enough for us to illustrate the various EDRM/litigation steps for purposes of the Search Project's work.

- A series of emails attaching a variety of ESI – other emails with attached documents, zipped folders containing numerous emails and attachments, and a range of other loose ESI (Word documents, Excel spreadsheets etc.).
- Three DVDs containing largely (but not completely) overlapping sets of the ESI collected during the investigation.

The ESI in the emails and on the DVDs does not appear to be organized in any structured manner, nor does it include information about custodians or source folders/machines. Ivan Isis did not include, and as far as Paula can determine has not prepared, any additional record of what data was collected, from what sources, using what technologies, or any other information about the ESI collection process.

Paula needs to coordinate a review and analysis of the ESI received from Ivan Isis. Among other things, she needs to create a record of what ESI she has received, what ESI she has reviewed (including her team's use of search queries), and what her assessments are.

Paula and her team retain an e-discovery vendor to help them analyze and review the ESI received from Isis LLP.

Processing & Analysis

Since the ESI has already been collected and the team does wish to perform a formal multi-person review, the evaluation of ESI at this stage of the case comprises the Processing and Analysis stages of the EDRM model. The e-discovery vendor processes the various sources of ESI so that it is possible to analyze and view the data within an e-discovery software application that has search and analysis capabilities.

Processing involves extraction of email container files and archive files (e.g. zip files) so that each individual item is cataloged and its associated meta data is captured. Indexing is also performed in order to allow for searching the full text of the documents. At this point, Paula and the vendor agree that they will perform limited searching and filtering as part of processing, only deciding to exclude certain operating and application files which are identified as being part of the NIST list.

Once processing is complete, Paula asks one of Proux's associates, Patricia Polk to analyze the ESI. Patricia decides to perform some simple keyword searches including:

- MagicEye
- Betacorp or Beta Corp
- The names of all the known parties including Alan Baker, Alex Arnold and Ivan Isis.

After running each of these searches, Patricia reviews some of the documents that the application's search engine has ranked highly based on their relevance ranking algorithm. The documents returned by the above search criteria confirm the information gained from some interviews that employees also refer to this project as "Project Magic."

Patricia also uses the application's social network analysis capability which automatically performs searches in order to identify who was talking to whom. In particular, Patricia wants to find out who Alan has been emailing.

Patricia notices from some of the filters or faceted search capabilities that the data contains foreign languages and identifies some additional important email addresses.

Based on their review of this preliminary data, and several follow-up interviews of Alpha employees, they learn:

- The MedicEye project is referred to inside Alpha as "MedicEye" or "Project Magic" or "Magic".
- Alan Baker worked at Alpha from January 1, 2005 until May 1, 2008. His email address was bakera@alphatech.com. He also has a Yahoo account: ohnonono@yahoo.com.
- Work on the MedicEye project began on January 1, 2004.
- Other Alpha employees who work or worked on the MedicEye project include – but may not be limited to – the following:
    - Aileen Arvidson of the Business Development Department (arvidsona@ alphatech.com) (January 1, 2004 to date).
    - Anton Agnew, a software programmer in the Software Development Department (agnewa@ alphatech.com) (March 15, 2005 to date). He is sometimes referred to as "AA" or "A.A."
    - Audrey Astor of the Marketing Department (astora@ alphatech.com) (June 1, 2006 to date).
    - Aki Arata of the Business Development Department in the Tokyo office (arataa@alphatech.com) (September 1, 2005 to date). He does his work both in English and in Japanese.
- While he was still employed by Alpha, Alan Baker may have been in touch with several people at Beta:
    - Bonnie Benson, Beta's CTO (bonnie@betacorp.com).
    - Brad Boqin, head of Beta's Special Projects Group (brad_boqin@betacorp.com).
- Alex Arnold, General Counsel of Alpha, had some involvement in the internal investigation after Alan Baker moved from Alpha to Beta. His email address is arnolda@alphatech.com.

- Ivan Isis of Isis LLP, who carried out the internal investigation, may have corresponded by email with people inside Alpha. His email address is Ivan_Isis@IsisLLP.com.

5.3 Proux LLP Undertakes an Organized Collection and Analysis of Alpha's ESI (*i.e.*, a more traditional early case assessment)

Based on the findings of the internal investigation and the interviews and ESI analysis that Patricia has performed, Alpha decides to proceed further with the case. Paula drafts a demand letter to Beta. Alex and Paula also decide to undertake a formal collection and analysis of Alpha's ESI in order to perform a comprehensive early case assessment. Alpha wants additional information before deciding to file a complaint.

After further discussions with Alpha personnel, Paula and her team learn that Alpha's ESI includes the following:

- 1 Microsoft Exchange server.
- 1 Lotus Notes server.
- 20 unlabeled backup tapes suspected of including email backups from 1/1/00 – 12/31/07.
- A complete set of email backup tapes from 1/1/08 forward – daily, weekly and monthly.
- 2 eRoom sites.
- 5 software programmers in Palo Alto and Chicago (including Anton Agnew) with ThinkPads running XP Pro.
- Personnel in the Marketing Department and the Business Development Department with Dell laptops running XP Pro.
- Networked file servers in Palo Alto and Tokyo containing source code, planning documents, financial analyses and market studies in English and Japanese.
- 20 unlabeled backup tapes suspected of including Palo Alto and Tokyo server backups from 1/1/00 – 12/31/07.
- A complete set of Palo Alto and Tokyo server backup tapes from 1/1/08 forward – daily, weekly and monthly.

Using the preliminary facts gathered thus far, Paula wants to

A. identify, collect and preserve potentially relevant data, and

B. carry out searches to identify and review potentially relevant data and isolate potentially privileged communications and documents.

Identification, Preservation, Collection, Processing and Analysis

The evaluation of ESI at this stage of the case involves many more of the stages of the EDRM model. She recognizes that in almost every phase of this process search will be an essential element.

Identification – As part of identification, Paula and Patricia begin to formalize who from Alpha and Beta are custodians of data relevant to the case, what time period is critical to the case and what content on shared information stores might be relevant to the case. In order to identify the appropriate custodians, they use fielded search and/or social analysis on the ESI that was previously collected and analyzed and conduct additional interviews. They identify up to 30 likely custodians, of which 5 are designated as high priority. They conduct interviews with various Alpha IT personnel and identify the multiple sources of ESI listed above and the location of that data. For the backup tapes, they work with IT to conduct sampling to determine whether all of the ESI identified needs to be collected or whether it falls outside the scope of the case. In order to identify content on shared information stores including file servers and the eRoom sites, they develop a set of search terms including important people, topics and time frames. They use these to search both content and metadata, such as last modified date, on the file servers and eRoom sites. It is important that not only full documents are searched but also all associated metadata, which can show patterns of interaction and key points of activity.

Based on an analysis of metadata and the text searches against the eRoom data, they determine that the file servers contain relevant information but the eRoom sites were never used by Alan or anyone involved in the MagicEye development. The eRoom ESI is thus unlikely to contain information relevant to the case.

Preservation - Paula prepares and distributes a Litigation Hold letter to the relevant custodians and IT personnel to suspend destructive activities, isolate and protect potentially relevant data associated with the custodians or on relevant shared information stores. This preservation of data will create a static set that can be used moving forward for collection, processing and analysis of data.

Collection – In order to expedite the early case assessment and keep it within an agreed budget, the team decides to first collect the high priority custodians and ESI on the networked file servers that match a set of search terms and a date range. This collected data is copied to a dedicated server at Alpha and is submitted to the e-discovery Vendor for processing and additional analysis. The preservation set will be retained and returned to for additional search and collection activities once the selection criteria have been further refined.

Processing and Analysis – The procedure for processing remains the same as for the initial Early Case Assessment. Application and operating system files are excluded. Files within email container files and archive files (e.g. zip files) are extracted so that each individual item is cataloged and its associated metadata is captured. This data need to be formatted into a digital form that can be searched against in the next stage. It is also important at this stage that the search engine being used can support indexing and searching Japanese as well as English since key documents are composed in both languages. Any translation capabilities would also be important to use at this stage so that an English query could be translated to Japanese to search documents in that language. If anyone in question can speak Japanese they could have attempted to hide communication in this second language.

This is where she must begin to record all item-level metadata as it existed before moving to the processing stage. At this point, Paula is basically taking a snap shot of what the data looks like and identifying that at this point in time this is how the data looked like in relation to the queries executed against it. Here also, the exact date the query took place and the actual query term is recorded. These steps are all necessary in order to prove defensibility of the search criteria in court.

This is also the stage where entity extraction is best used. This means that specific heuristics are used to identify the people, places and things that are re-occurring within the collection set. It is also time to run de-duplication so that only a single instance of a document is processed for analysis. This is another stage where unrelated data can be removed from the collection set. For example, in the collection stage all emails sent and received in a specific date range may have been collected. Using search methods such as key terms or analysis tools like clustering may show which email is spam. Search criteria may then be developed to exclude these types of documents to ensure they do not go to the next stages of analysis or ultimately to review.

At this point in the project, the goal of this phase is not to collect all the data that are relevant to these key terms and individuals but rather to continue to identify a list of good key terms or fielded searches that can be used for the additional collection and processing needed once production requests are received and also to gather information regarding the issues of the case and the documents supporting same. The searches being developed can then be used either during the collection or processing phases to reduce the volume of data that must be reviewed by the case team.

There are several types of advanced search methods that can be used to identify hidden terms, unknown relationships or suggested other searches. These include such things as conceptual search, clustering and auto-classification. Conceptual search will identify related terms and bring searches back for those items. So perhaps through conceptual search they would identify other documents that had related terms. Clustering will automatically categorize result sets into hierarchical categories to demonstrate hidden relationships or identify groups of documents that may be irrelevant to the case. Then finally, using auto-classification all documents could be categorized again looking for hidden themes or eliminating irrelevant documents. All advanced search methods will help identify key terms or document types that may be used for additional collection and processing.

In addition to the advanced search technologies described above, Patricia also performs more traditional full text searches. Understanding that the project name is 'MedicEye' and the other nick name is 'Project Magic' is important. Using the same methodology for discovery as outlined above is critical for identifying additional keyword terms. Paula begins scanning documents and performing some general queries to see if the project has other code names. She also will be looking to see if any of the custodians have other names that they are referenced as or reference themselves as. Additional searches may be performed against multiple fields. An example is looking for emails sent by one custodian to another custodian within a specific date range containing a key term in the title. This is helpful to identify very specific document sets. It is also important to execute queries using a variety of query operators. This stage is helpful to do queries such as find "patent" within 8 words of "Project Magic" to find documents that have both terms in close proximity of another. Another, query type could be a wildcard to find cases where perhaps a misspelling has occurred. Detection of misspellings can be performed in an automated way or via manual searches, such as "Ma*c". The query would find documents that have 'Magic', 'Majic', 'Magc' , etc. Care would need to be taken to exclude false positives such as magnetic. When typing quickly, letters are often left out of words or incorrectly typed. By testing these search criteria, Paula and Patricia can determine how effective their search criteria are in returning responsive documents.

The analysis stage is also the point where Paula must ensure that all assumptions are true in identifying critical metadata. For example, she must know that when searching exchange bcc's are not always indexed and require specific mechanisms to extract this kind of data. She must also know that metadata used to describe content in eRoom might be stored in a separate system all together. It is critical that in the analysis stage all of these atypical behaviors are recognized and accounted for. This stage alone may take Paula the most amount of time to navigate through everything and begin building the case she needs in preparation of review.

Throughout this iterative process, the queries performed and the associated documents resulting from the particular queries should be captured and recorded to assist Paula and Patricia with the development of their searches but also to add to the defensibility of their process.

Once they complete their analysis above, Paula and Patricia develop and document an initial list of textual search terms, fielded search terms (like date range) and any other search methodologies that they would like to be used in the case going forward.

5.4 Beta Begins Preparations for a Possible Lawsuit after Receiving a Demand Letter from Alpha

Beta's General Counsel, Bettina Bloch, receives a letter from Alpha's outside counsel, Paula Proux, which describes Alpha's understanding of the general facts of Alan Baker's departure from Alpha and advises Beta that Alpha intends to take "whatever legal steps are available and necessary" to protect Alpha's rights and interests.

Bettina Bloch meets with former Alpha employee Alan Baker, Beta's CEO and several other senior Beta personnel, and with Diego Dominguez of Dominguez LLP, Beta's regular outside counsel, to gather some preliminary facts and discuss how to respond to Alpha's letter. Among other things, Bettina and Diego identify Beta employees who may have information, hard-copy documents or ESI regarding the issues raised by Alpha's letter and they prepare and distribute a litigation hold notice to those Beta employees.

Based on their preliminary meetings, interviews and other inquiries, Bettina and Diego learn that –

➢ The relevant facts include:

- The Beta project intended to compete with Alpha's MedicEye project is referred to inside Beta as "InsureTrak", and sometimes as "Project AlphaBust".
- Alan Baker has worked at Beta since May 15, 2008. His Beta email address is bakera@betacorp.com. As Alpha learned, he also has a Yahoo account: ohnonono@yahoo.com.
- Work on the InsureTrak project began on January 1, 2007.
- Other Beta employees who work or worked on the InsureTrak project include – but may not be limited to – the following:
    - Bonnie Benson, Beta's CTO (bonnie@betacorp.com) (June 15, 2006 to date).
    - Brad Boqin, head of Beta's Special Projects Group (brad_boqin@betacorp.com) (September 1, 2005 to date).
    - Bashir Baktiar, a software developer in Beta's Abu Dhabi office (bashir@betacorp.com) (November 1, 2004 to date). He does his work both in English and in Arabic.
    - Boris Bonaparte, a software developer in Beta's Paris office (boris@betacorp.com) (April 1, 2006 to date). He is sometimes referred to as "Napoleon". He does his work both in English and in French. He also maintains a set of Beta servers, located in the

Paris office, which contain source code, personnel records (including Alan Baker's), and Beta's email (Microsoft Exchange).

- Bettina Bloch, General Counsel of Beta, had some involvement in the recruitment of Alan Baker from Alpha. Her email address is bettina_bloch@betacorp.com.
- Diego Dominguez of Dominguez LLP, Beta's outside counsel, was also consulted in connection with Beta's recruitment of Alan Baker. His email address is diego@dominguezesq.com.

➢ Beta's ESI includes the following:

- 1 Microsoft Exchange server (located in the Paris office).
- A complete set of email backup tapes from 1/1/03 forward – daily, weekly and monthly.
- 7 software programmers in Boston, Paris (including Boris Bonaparte) and Abu Dhabi (including Bashir Baktiar) with ThinkPads running XP Pro.
- Networked servers in Boston and Paris containing source code, planning documents, financial analyses and market studies in English and French.
- A complete set of Boston and Paris server backup tapes from 1/1/08 forward – daily, weekly and monthly.

Using the preliminary facts gathered thus far, Diego Dominguez and his team want to

A. identify, collect and preserve potentially relevant data, and

B. carry out searches to identify and review potentially relevant data, including isolating potentially privileged communications and documents.

Identification, Preservation, Collection, Processing and Analysis

The evaluation of ESI by Diego and his team also involves many more of the stages of the EDRM model.

Identification – Following their internal interviews, Diego and Bettina have developed an initial list of Beta employees that may have data relevant to the case, the critical time period and what content might be relevant.

Preservation - Diego prepares and distributes the Litigation Hold letter to suspend destructive activities, isolate and protect potentially relevant data. This preservation of data will create a static set that can be used moving forward for collection, processing and analysis of data.

Collection – Various search, sampling and analysis techniques may be employed against the preserved data set to analyze the documents that they have available regarding the facts at hand and also to begin the development of search criteria that they may use moving forward in the case.

As Alpha had previously done, Diego used fielded search and/or social analysis, to recognize who at Beta was in contact with and could present relevant details to the case. Diego also performed similar searches for Bonnie Benson, Brad Boqin, Bashir Baktiar, Boris Bonaparte and Bettina Bloch as they were identified as persons with knowledge of the issues and potential custodians of relevant ESI.

Diego also created searches to identify the date ranges of emails sent during this time in question, documents created and even documents edited based on last modified date and time.

Once Diego developed an initial scope for the project, he contacted and hired his own e-discovery vendor and submitted the collected data for processing and additional analysis.

Processing and Analysis – The ESI collected for the preliminary group of custodians from the Beta's much larger preserved data set is sent to their e-discovery Vendor for processing. The preservation set will be retained and returned to for additional search and collection activities once the selection criteria have been further refined.

The procedures and issues discussed above relative to processing are the same for Beta as they were for Alpha's case assessment. Beta now needs to identify and develop their own list of good key terms or fielded searches that can be used for the additional collection and processing needed once production requests are received and also to gather information regarding the issues of the case and the documents supporting same. The searches being developed can then be used either during the collection or processing phases to reduce the volume of data that must be reviewed by the case team.

They used similar types of advanced search and full text searches as described above only targeted to their particular issues. For example, Diego created searches for “InsureTrak”, “Project AlphaBust” and “AlphaBust” and upon reviewing documents responsive to those queries may make further refinements to his search criteria.

Throughout this iterative process, the queries performed and the associated documents resulting from the particular queries should be captured and recorded to assist Diego with the development of his searches but also to add to the defensibility of his process.

5.5 After Alpha Sues Beta in Federal Court (D. Md.), Outside Counsel for Alpha and Beta Prepare for and Engage in Rule 26(f) Meet-and-Confer Process

Outside Counsel should first familiarize themselves with the current case law and any local rules applicable to the discovery of ESI. In this case, the District Court of Maryland has a “Suggested Protocol for Discovery of Electronically Stored Information (“ESI”).”

A. Preparing For the Rule 26(f) Meet And Confer

In preparing for the Rule 26(f), the goals of Outside Counsel for Alpha and Beta should be to:

(1) define the sources of and become familiar with potentially responsive information (*i.e.* systems, servers, custodians and documents);

(2) understand their client’s ESI policies related to those sources of potential responsive information;

(3) instruct their clients and the identified custodians about their preservation (*i.e.* “litigation hold”) obligations; and

(4) determine how to identify responsive information (*i.e.* time restrictions, search methods, search terms, search results), including running preliminary searches to test adequacy of suggested approach.

(i) Alpha

Potential Sources of ESI. During its investigation, Proux identified the potential sources of ESI within Alpha as detailed above in Section 5.3.

Jurisdictional Issues. Potentially Responsive ESI may reside on networked servers in Japan. This raises jurisdictional issues that may come into play during discovery. Proux should begin the research and analysis of international laws and treaties applicable to obtaining this information within a foreign sovereign territory.

Document Retention Policies Related to Sources of ESI. Proux next needs to determine the back-up, retention and destruction policies related to this ESI. Questions to ask:

- When was the policy implemented?
- How is the policy enforced?
- Has the policy changed during the relevant time period?

Proux's investigation revealed that Alpha changed its backup tapes for both its email and networked servers on January 1, 2008. Proux needs to understand why this change was made and if the policy changed at the same time as well and, if so, how it changed.

Become Familiar with Current and Past ESI. Proux needs to gain an understanding, usually through the assistance of an Alpha employee(s), of how each source of ESI works, how it used by Alpha employees and which Alpha employees have access to this ESI.

Identify Key Witnesses and Key Custodians. Proux also identified the following Alpha employees that may either have responsive ESI related to the case (*i.e.* witnesses) or may be Alpha representatives that can assist Proux in understanding and accessing responsive ESI (*i.e.* custodians):

- Alan Barker, former employee, Software Development Department
- Aileen Arvidson, Business Development Department
- Anton Agnew, Software Development Department
- Audrey Astor, Marketing Department
- Aki Arata, Business Development Department, Tokyo Office
- Alex Arnold, General Counsel of Alpha (privileged)
- Ivan Isis, Outside Counsel, Isis LLP (privileged)
- Information Systems Custodian, Information Technology Custodian, Network Administration Custodian, Records Management Personnel

Suspension of Document Retention Policy and Litigation Hold. The scope of the suspension and hold will be based on the particular facts of a case. Here, Proux should suspend any overwrite, deletion and/or purging related to systems and servers identified in the investigation. The suspension and hold notifications should extend to all the ESI Proux identified in its investigation and should issue to the witnesses and custodians identified by Proux in its investigation.

Someone within Alpha legal should be designated as the litigation hold compliance officer who monitors compliance and who sends periodic reminders. Additionally, as the case develops the suspension and hold notices should be revisited both as to their scope as well as whether additional witnesses need to be added.

Preliminary Searches.

> Based upon the work that Paula and her team performed during their case assessment and the documentation created during the criteria development process, they shared with Beta what they felt were search criteria that could be effectively used to reduce the volume of data needed to be reviewed while returning relevant material.

(ii) Beta

Potential Sources of ESI. During its investigation, Dominguez identified the potential sources of ESI within Beta as detailed above in Section 5.4.

Jurisdictional Issues. Potentially Responsive ESI may reside on networked servers in Paris and Abu Dhabi. This raises jurisdictional issues that may come into play during discovery. Dominguez, like Proux, should begin the research and analysis of international laws and treaties applicable to obtaining this information within a foreign sovereign territory. Since both parties have jurisdictional issues, this issue should definitely be raised at the Rule 26(f) meet and confer.

Document Retention Policies Related to Sources of ESI. Dominguez next needs to determine the back-up, retention and destruction policies related to this ESI. Questions to ask:

- When was the policy implemented?
- How is the policy enforced?
- Has the policy changed during the relevant time period?

Dominguez's investigation revealed that Beta only has backup tapes for emails servers from January 1, 2003 and networked servers from January 1, 2008. This should not be a problem given the probable relevant time frame Beta may have obtained Alpha's confidential information. However, Dominguez needs to understand the history of the email and network servers, including whether any information that pre-dates the back-up tapes is archived or stored in any manner.

Become Familiar with Current and Past ESI. Dominguez needs to gain an understanding, usually through the assistance of a Beta employee(s), of how each source of ESI works, how it used by Beta employees and which Beta employees have access to this ESI.

Identify Key Witnesses and Key Custodians. Dominguez also identified the following Beta employees that may either have responsive ESI related to the case (*i.e.* witnesses) or may be Beta representatives that can assist Dominguez in understanding and accessing responsive ESI (*i.e.* custodians):

- Bonnie Benson, CTO
- Brad Boqin, Special Projects Group

- Bashir Baktiar, Software Developer in Abu Dhabi Office
- Boris Bonaparte, Software Developer in Paris office
  (*Note*: Bonaparte is the custodian of email and network servers; he may also be a witness in the case. Dominguez should consider whether or not to suspend Mr. Bonaparte access to these servers during the pendency of the case).
- Bettina Bloch, General Counsel (privileged)
- Diego Dominguez, Outside Counsel, Dominguez LLP (privileged)
- Information Systems Custodian, Information Technology Custodian, Network Administration Custodian, Records Management Personnel

Suspension of Document Retention Policy and Litigation Hold. The scope of the suspension and hold will be based on the particular facts of a case. Here, Dominguez should suspend any overwrite, deletion and/or purging related to systems and servers identified in the investigation. The suspension and hold notifications should extend to all the ESI Dominguez identified in its investigation and should issue to the witnesses and custodians identified by Dominguez in its investigation.

Someone within Beta legal should be designated as the litigation hold compliance officer who monitors compliance and who sends periodic reminders. Additionally, as the case develops the suspension and hold notices should be revisited both as to their scope as well as whether additional witnesses need to be added.

Preliminary Searches.

Based upon the work that Diego performed during his case assessment and the documentation created during the criteria development process, he also had developed a list of search criteria that he felt should be performed.

B. The Rule 26(f) Meet And Confer

Proux and Dominguez have engaged in a thorough early case assessment and preparation and are now ready for a meaningful Rule 26(f) conference to set the stage for discovery in this matter.

The primary goals of the meet and confer process should be to:

(1) agree to terms of appropriate protective order;

(2) agree to technical form of production, including provisions for native files, static files, metadata, redaction and identifying labeling such as bates-stamping;

(3) possible exchange of sources of ESI, including technical specifications and identification of ESI "not reasonably accessible" and potential cost-shifting, discuss scope of suspension of document retention and litigation hold; and

(4) agree on search methodologies and search terms.

Protective Orders. The parties should discuss the terms of a protective order given that this matter concerns Alpha's trade secrets. One of the most critical provisions concerns the protocol for handling the inadvertent production of privileged information. Rule 502 of the Federal Rules of Evidence now provides statutory protections against the inadvertent production of ESI and should be considered when drafting this provision of the protective order. Additionally, if meta-data is going to be produced, the parties should discuss how to handle, including an agreement not to view meta-data or specific types of meta-data without notice to the producing party.

Technical Form Of Production And Limits On Initial Scope Of Discovery. The parties should agree to whether production will be static (.pdf, .tiff) image or native image. If native files will be produced, the parties need to consider how to handle meta-data and redactions of meta-data or native files. The parties need to discuss the method of identifying documents produced (*e.g.* Bates stamping or renaming native files with a numbering system).

The parties also need to consider the scope of discovery requests, including an initial focused stage of discovery. For example, the parties might limit their initial search of ESI to that which is reasonably accessible. This might mean initially searching (a) only the on-line email server and network server; or (b) the local drives and email accounts of the witnesses identified in Proux and Dominguez investigations; or (c) only the complete sets of backup email and network tapes from January 1, 2008 forward.

Sources of ESI, Hold PolicyaAnd ESI "Not Reasonably Accessible." Proux and Dominguez might agree to exchange the technical details of Alpha's and Beta's information and network systems that might contain responsive ESI. The parties could also exchange the key custodians identified in their respective investigations. The parties should exchange each other's litigation hold policy in order to identify any objections early-on to what information is, or is not, being preserved. The parties might also consider preliminary custodian of record depositions related to the collection of ESI.

The parties should also discuss the types of ESI that might be "not reasonably accessible" and any associated costs related to the preservation, retrieval and production of ESI "not reasonable accessible." For example, Alpha has unlabeled backup tapes from both email and network servers that might contain responsive ESI. The Beta investigation revealed that it only has server backup tapes from January 1, 2008 forward. Alpha may want to inquire about the accessibility of server ESI that pre-dates 2008.

Search Methodologies and Search Terms. The parties should discuss and agree on proposed search methodologies (*e.g.* keyword, Boolean, concept). Restrictions and limitations on the search methods, documents searched and time frames should be discussed. A set of common keywords could be proposed by each side and agreed upon. The parties could also agree to an initial sampling of ESI returned from the common search terms.

Both parties exchanged and discussed their initial criteria that each side had developed during their case assessment process.

Both parties agreed upon a list of custodians to be collected and also the date ranges to be applied. A method of sampling was also discussed relative to the many backup tapes that both parties had within their ESI. It was felt that there would be a large amount of duplication would be contained within the sets but that sampling would be performed to determine the volume that needed to be restored and collected from the preservation sets.

Each side also had suggested changes to the search criteria to be used and agreed to run the modifications against the sample data that they had already processed to provide them with an opportunity to review the requested changes and to determine the effectiveness of those changes.

During this sampling exercise, Alpha found that several of Beta's requested terms contained wildcards that returned an overly broad set of results that did not return a high percentage of responsive documents. They then provided alternate search criteria created to address the issue and were able to negotiate an agreed upon list of terms with Beta with the understanding that as additional data was processed and anomalies were found within the results that either side could request additional negotiation of the search criteria.

Also, both sides requested that a sample of documents be reviewed that are not hitting on any of the search criteria to determine whether any of those documents are responsive and, if so, additional criteria will need to be developed to capture those items.

5.6 Alpha and Beta Run Searches, Propound Discovery And Produce ESI

Based upon the parameters established in the Rule 26(f), Alpha and Beta both propound discovery requests. Each company uses the agreed upon searches to review results and produce ESI.

Collection, Processing, Analysis, Review and Production

The evaluation of ESI at this stage of the case involves different stages of the EDRM model as well as additional iterations through many of the stages already discussed. Identification and Preservation were already completed by both parties during their initial case assessments.

Collection – Although they had collected certain data sets to be used with their case assessment processes, there was now a larger volume of data that needed to be collected for processing due to the agreed upon searches and the discovery requests.

Processing – The mechanics of processing, particularly as they relate to searching, have already been discussed during the Alpha and Beta case assessments in Sections 5.3 and 5.4.

Now that the case is entering a much larger scale review process, classification criteria may be used by either of the case teams to organize the documents in review to assist with the review process. One of the most common types of classification criteria are those searches developed to identify potentially privileged documents. Attorney names, law firm domains and other common terms may be used to identify these documents. Exclusionary criteria may also be used if a privilege term is found to be overly broad. For example, if the term "privileged and confidential" were to be used it is quite often found in email footer language and the search term may need to be modified to exclude those items where the term is found only in the footer but capturing other instances of the phrase.

Analysis & Review – In addition to the use of classification criteria, many of the same advanced searching methodologies that were discussed earlier (e.g. conceptual search, clustering and auto-classification) may be used during the review process to assist reviewers in making consistent review calls across the larger review set.

Also, during review feedback relative to search criteria should be provided to the case team. The reviewers may find that certain searches are yielding a large volume of non-responsive documents. Samples of the documents should be reviewed and modifications to the criteria may be made to limit those types of documents.

Additionally, reviewers may find new issues that need to be addressed by search criteria that are not contained within the current search criteria lists.

Production – once the review completes, or as agreed upon during the 26(f) productions will be made to the opposing side.

*   *   *

Use Case Scenario – Actors

Plaintiff	Defendant	Unaffiliated Third Parties
Party: Alpha Corporation **Current employees:** Anton Agnew (formerly James Winston) Aki Arata (formerly Yoshi Tanaku) Alex Arnold (General Counsel) (formerly Larry Landry) Aileen Arvidson (formerly Susan Smith) Audrey Astor (formerly Ellen Eleanor) **Alpha's outside counsel:** Proux LLP (formerly Law Firm 1) Paula Proux, Partner Patricia Polk, Associate **Counsel for Audit Committee investigation:** Isis LLP (formerly Law Firm 2) Ivan Isis	**Party:** Beta Corporation **Current employees:** Alan Baker (former Alpha employee) (formerly Alan Adams) Bashir Baktiar Bonnie Benson (formerly Rona Roberts) Bettina Bloch (General Counsel) Brad Boqin (formerly Joe Liu) Boris Bonaparte **Beta's outside counsel:** Dominguez LLP Diego Dominguez	

6. Search Methodologies

Search in the EDRM context can mean employing any of a number of techniques across a variety of data. Usually the objects being searched are documents in a case, but even there the documents can take several forms. The proliferation of computer files (known as Electronically Stored Information, or ESI), whether common files like documents created with Microsoft Word© or PowerPoint©, email stored as individual message files or together in an Outlook or Notes data file, OCR files created from scanned paper documents, or even more exotic files such as those created by a CADCAM program, have caused the need for larger computer systems to store and manage the data in these. Also, the text within a document is not the only data that can be searched. Information about the documents themselves, known as *metadata*, can also be stored (usually in a database) and searched. This section discusses known search techniques used by existing computer systems to search the data available.

Search tools and methodologies have numerous applications during the e-discovery phase of the litigation lifecycle. The following provides a real life example of the processes and challenges related to using search and how these challenges can be mitigated.

Attorney James Smith is working on a new case involving a motorcycle accident. The plaintiff is claiming that his local garage failed to spot the leaking brake fluid from his BMW 1150 motorcycle when he was at the garage for maintenance, which caused a mechanical failure leading to the accident. Attorney Smith, who is representing the defendant garage, has a database containing thousands of documents, including email to and from the plaintiff and the defendant, email from a mailing list for motorcycle aficionados that both plaintiff and defendant participated in, and OCR'd documents including maintenance records and receipts from the garage.

Attorney Smith wishes to find the responsive documents as quickly as possible, without having to read each of the thousands of documents in the database. In addition, he wants to be sure he does not fail to mark any privileged documents appropriately. The database has a search feature, so he decides to use it. Being familiar with Google, he enters the following search terms

motorcycle maintenance records

This type of search, using a list of key words, is known as a *keyword search*. The set of documents that are returned all contain at least one of the words in his list of search terms. This is his *result set*.

The search engine that Attorney Smith is using highlights the words that matched his search when he looks at the documents. He notices that the results contain maintenance records for automobiles as well as motorcycles, and in a couple of cases contained email about music, not maintenance, records. He changes his search terms to:

motorcycle AND “maintenance records”

Putting in the quotation marks means that his result set will only contain matches for the entire phrase contained within the quotation marks, not for the individual words. When he adds the word AND into his search query, he created what is known as a *Boolean search*. This means a search contains the words AND, OR, or NOT, and tells the search engine more about what you mean when you enter your terms. In Attorney Smith's case, this means his result set will contain documents that have both the word *motorcycle* in them as well as the phrase *“maintenance records”*.

The documents in this result set have maintenance records for all sorts of different kinds of motorcycles, so Attorney Smith changes his search terms to be more specific. He also limits his search so that the search will only run against the current subset of the documents. This is called a *subset search*. This time he enters:

“BMW 1150” AND brakes

Attorney Smith realizes from looking at previous documents that entering just "brakes" won't cover enough options, so he turns on the *stemming* option and runs his search again. Having stemming turned on means that not only documents containing the word *brakes* will match his search, but also documents containing *brake*, *braking*, and *braked*.

Attorney Smith is pleased with the results he obtains from this search, but he soon realizes that he doesn't see any of the actual maintenance records from his client's garage. These were paper copies, so they were first scanned, then run through an OCR process to convert the text into a machine-readable form, before being loaded into the database for searching. The problem with OCR text taken from a scanned document is that sometimes the OCR misses a letter or two, so searches might not match.

Attorney Smith runs his search again, this time turning on the *fuzzy search* option. Fuzzy search allows a search term to match other terms that don't match exactly, but might be off by a letter or two. This way, if the word *brakes* was misread as *brokes* or even *blokes*, it would still match the search term.

Finally, Attorney Smith has a collection of documents that contains what he needs, and he is ready to prepare review sets for his team to begin reviewing. However, he decides to run one more type of search just to be sure. This search is called a *concept search*, and it uses statistics to match not just exact keywords but to match concepts that are similar to the keywords entered. Concept search can also be used without keywords, by simply finding all the major concepts and how they are clustered within the set of documents.

This time Attorney Smith runs a concept search using the keywords *BMW 1150*, *brakes*, *accident*, and *maintenance*. As he scrolls through the results he doesn't see anything new, until he sees the word *stoppies*, which he is unfamiliar with. A little digging in the result set of documents lets him discover that stoppies is a behavior similar to wheelies that can result in damaged brakes. The documents containing this word revealed that the plaintiff frequently engaged in this dangerous behavior. Attorney Smith now had the ammunition he needed to win his case, using a concept he did not know in advance existed.

6.1 Keyword Search

A *keyword search* is a basic search technique that involves searching for one or more words within a collection of documents. Typically, a keyword search involves a user typing their search request, or query, into a search engine such as Google, which then returns only those documents that contain the search terms entered. The documents returned by the search engine are called the *search results*.

6.1.1 Guidelines

Keyword searches are most often used to identify documents that are either responsive or privileged. It is also widely used for large-scale culling and filtering of documents. Keywords often form a basic building block for constructing other more complex compound searches. Such compound searches use other search elements such as Boolean logic.

6.1.2 Normal Parameters

Keyword search normal parameters are:

1. The syntax in the search string;
2. Use of the keywords with or without stemming;
3. Use of keywords with certain wildcard specifications and the syntax for said wildcards;
4. Case-sensitivity of keywords used in searches and whether the keyword should match both cases; and
5. The target data sources to be searched.
    a. Whether the query can be applied to any specific fields such as email 'To/From' or 'Subject'.
    b. Whether the query can be applied to any specific date range such as an email 'Sent Date' between the date range of January 1, 2001 through December 31, 2001.

6.1.3 Assumed Parameters

The examples used in the Search Guide are all given in standard U.S. English. However, some documents may be written in non-English languages or character sets, and the following parameters may need to be specified. These are discussed in further detail in Section 7.

1. The character encoding of the text – UTF-8, UTF-16, CP1252, Unicode/WideChar etc.
2. Language of the keyword, to select appropriate stemming.
3. Any special handling of characters such as diacritics, accents etc.
4. If de-compounding of the keyword needs to be performed, usually when working with languages such as German.
5. If there is a set of special characters, what the special characters are, and how an escape character is specified.
6. If there is a tokenization scheme present, what the token delimiters are, and the impact of tokenization on searchability of documents. Searches may not be precise when these tokenization characters are present in the keyword.

6.1.4 Phrase Search

Keyword searches also encompass phrase searches. Phrase searches can be specified by enclosing the words in the phrase between two quotation marks (").

The following should be considered when conducting a phrase search:

a) Phrases that contain a double-quote in any of its keywords require escaping of the double-quote. Example: to search for the quoted expression

    He said: "don't do it"

the query would be:

“He said: \”don’t do it\””

b) If there are *noise words* (see below), they will be specified in the phrase. However, the searching implementation may substitute any word in place of the noise word. Therefore, it is very important for the legal practitioner to validate the behavior of phrase searches that include noise words.

c) If the phrase includes special words (such as the Boolean Operator AND), they must be enclosed within double-quotes so they will be interpreted as part of the phrase and not as an operator.

6.1.5 Wildcard Specifications

A legal professional can specify a single character wildcard or a multiple-character wildcard in the following ways:

Wildcard type	Syntax	Description
Single-character wildcard	x?y	matches all strings that begin with substring x, end with substring y, and have exactly one-character in between x and y
Multiple-character wildcard	x*y	matches all strings that begin with substring x, end with substring y, and have 0 or more characters between x and y

If a keyword contains wildcards, an escaping mechanism is needed to search. To escape, the following syntax should be used:

x\?y – search for a string with an embedded string.

For example: to search for

“How are you?”

the search string would be:

“How are you\?”

Availability of multi-character wildcards may be limited in some systems. Some search engines require a certain number of leading characters and do not support search terms that start with a wildcard.

6.1.6 Truncation Specification

Truncation specification is one way to match word variations. Truncation allows for the final few characters to be left unspecified.

Truncation is specified using the following syntax:

Syntax	Description
x!	matches all strings that begin with substring x.
!x	matches all strings that end with substring x.
“x! y”	when specified within a phrase, the truncated match on the words with !, and exact match on the others.

6.1.7 Stemming Specifications

Stemming specification is another method for matching word variations. Stemming is the process of finding the root form of a word. The stemming specification will match all morphological inflections of the word, so that if you enter the search term *sing*, the stemming matches would include *singing*, *sang*, and *song*. Note that even though a stemming search will return *singing* for a search term of *sing*, this is different from wildcard search. A wildcard search for *sing** will not return *sang* or *song*, while it will return *Singsing*.

Syntax	Description
x~	matches all morphological variations (inflections) of the word. Exactly how a search implementation identifies these inflections is not specified.
“x~ y”	when specified within a phrase, the stemming variations match on the words with ~, and exact match on the others.

6.1.8 Fuzzy Search

Fuzzy search allows searching for word variations such as in the case of misspellings. Typically, such searching includes some form of distance and score computations between the specified word and the words in the corpus.

Fuzzy search is specified using the operator: fuzzy-search.

Syntax	Description
fuzzy-search(x,s)	For the search word x, find fuzzy variations that are within the score s. The score is specified as a value from 0.0 to 1.0, with values closer to 1.0 being a closer match. The word itself, if present, will match with a score of 1.0.
fuzzy-search(x,s,n)	For the search word x, find fuzzy variations that are within the score s and limit the results to the top n by score.

Fuzzy search may be combined with other search constructs, and be included as part of a phrase or Boolean constructs.

6.1.9 Errors to Avoid

Some caveats when using keyword searches are:

1. Stemming may cause additional unintended keyword matches.
2. Wildcard expansions may cause results to be overly broad.
3. If tokenization is based on certain text characters being interpreted as delimiters, they may not be searchable as a keyword. Consider using a phrase as a search.
4. Case-sensitivity may need to be considered carefully.
5. If a word in a document contains a hyphen and the keyword matches any or all of the hyphenated word, depending on how the document is indexed the hyphen may prevent a match. For example, if the keyword is *known* and the document contains *well-known*, there is a chance that the search engine will not recognize the two as a match.
6. If the document is structured as a compound document (i.e., has multiple sections such as Title, Body etc.), keyword-based searches should be performed with care.

6.2 Boolean Search

Boolean searches are used to combine results of multiple searches as well as to designate ambiguity, as when search for two or more terms but do not necessarily need both. They are specified using Boolean operators as shown below:

Operator	Description
AND	This is specified between two keywords and/or phrases, and specifies that both of the items be present for the expression to match.
OR	This is specified between two keywords and/or phrases, and specifies that either of the two items be present for the expression to match.
NOT	Negates the truth value of the expression specified after the NOT operator.
NOT w/n	Specifies that the terms and/or phrases to the right of the w/n specification must not be present within the specified number of words.
ANDANY	This is specified between two keywords and/or phrases, and specifies that items following the ANDANY operator are optional.
w/n	Connects keywords and/or phrases by using a nearness or proximity specification. The specification states that the two words and/or phrases are within n words of each other, and the two words/phrases can be in either order.

	NOTE: the specified number of words implies that there are n-1 intervening other words between the two. “Noise words” are counted in the specification.
pre/n	Connects keywords and/or phrases by using a nearness or proximity specification. The specification states that the two words and/or phrases are within n words of each other, and the order of the words is important.
w/para	The two keywords and/or phrases are found within the same paragraph, and order is not important.
pre/para	The two keywords and/or phrases are found within the same paragraph, and order is important.
w/sent	The two keywords and/or phrases are found within the same sentence, and order is not important.
pre/sent	The two keywords and/or phrases are found within the same sentence, and order is important.
start/n	The keyword/phrase is present at the start of the document or section, within n words of the start.
end/n	The keyword/phrase is present at the end of the document or section, within n words of the end.

6.2.1 Errors to avoid

There are several issues to consider when using Boolean Search.

6.2.1.1 Evaluation ambiguity

Ambiguous evaluation of operators occurs when the operators are specified without understanding the order of evaluation. For example,

owners AND dogs OR cats

If the intent is to find documents containing pet-owners and either dogs or cats, the above search string could produce inaccurate or unexpected results. To avoid this, use grouping operators, that is, use parenthesizes to enclose search terms that should be evaluated at the same time:

owners AND (dogs OR cats).

6.2.1.2 Effect of Document Segments

In some situations, a document is split into multiple segments (such as Abstract, Body, Title, References, Citation, etc.). In these situations, the Boolean operators may be limited to a specific document segment. In these situations, you may need to specify the search scope of the document.

6.2.1.3 Evaluation Order

Although the evaluation order should be immaterial, you may find that some search engines produce different results if the order is specified differently. As an example, “cats AND dogs” should produce the same results as “dogs AND cats”. In other implementations, the performance of search is impacted by the order of specification. As an example, “owners AND (cats OR dogs)” performs better (produces search results faster) than “(cats OR dogs) AND owners”.

6.2.1.4 Boolean Operators as Keywords

When Boolean Search operators are themselves searched either as a keyword or as part of a phrase, care should be taken to avoid them being interpreted as Search Strings. To specify keywords that would otherwise be interpreted as a Boolean Search Operator, the keywords can be enclosed within double quotes.

Example: to search whether a document contains the string w/5 within it, specify the query by “w/5”.

6.3 Grouping

Grouping is used to specify the precise order of evaluation of Boolean Search Constructs. This is achieved using parenthesized constructs as shown below:

Syntax	Description
((A OR B) AND (B OR C))	Grouping by parenthesis allows individual expressions to be evaluated per the parenthesis.

The only grouping characters are ‘(‘ and ‘)’. These are meta-characters and must be escaped if they should be searched for. As an example, to search for the phrase: “Contract (Sales Department)”, specify: “Contract \(Sales Department\)”

6.4 Synonym Search

Synonyms are word variations that are determined to be synonyms of the word being searched. Such searching includes some form of dictionary or thesaurus based lookup.

Synonym search is specified using the operator: synonym-search.

Syntax	Description
synonym-search(x)	For the search word x, find synonym variations.

Synonym search may be combined with other search constructs. Synonym search may also be included as part of a phrase or Boolean construct.

6.5 Related Words Search

Related words search allows a legal professional to specify a word and other words that are deemed to be related to it. Typically, such related words are determined as either part of concept search or by statistical co-occurrence with other words.

Related word search is specified using the operator *related-word-search*.

Syntax	Description
related-word-search(x)	For the search word x, find other related words.

Related words search may be combined with other search constructs, and be included as part of a phrase, or Boolean constructs.

6.6 Concept Search

Concept search allows a legal professional to specify a concept and documents that describe that concept to be returned as the search results. It can be a useful technique to identify potentially relevant documents when a set of keywords are not known in advance. Concept search solutions rely on sophisticated algorithms to evaluate whether a certain set of documents match a concept. There are three broad categories of concept search that a legal practitioner may need to understand and evaluate its applicability.

6.6.1 Latent Semantic Indexing

Latent semantic indexing (sometimes also referred to as Latent Semantic Analysis [9]) is a technology that analyzes co-occurrence of keyword terms in the document collection. In textual documents, keywords exhibit *polysemy* (which refers to a single keyword having multiple meanings) as well as *synonymy* (which refers to multiple words having the same meaning). An additional factor is certain keywords are related to the concept in that they appear together. These relationships can be "is-a" relationship such as "motorcycle is a vehicle" or a containment relationship such as "wheels of a motorcycle".

In the "motorcycle" example above, documents may contain helmets, safety and brakes but not the word motorcycle. Additionally, the same document may also contain references to insurance, accident, the rider's name and a geographical location. Intuitively, a reader of the document may relate to this as a document on motorcycles, based on certain relevant terms while ignoring the presence of other irrelevant words. Latent semantic indexing understands the co-occurrence of these words, while reducing/eliminating the impact of other unrelated words.

6.6.2 Text Clustering

Text clustering is a technology that analyzes a document collection and organizes the documents into clusters. Refer to [10] for a review of various Text Clusteing approaches. This clustering is usually based on finding documents that are similar to each other based on words contained within it (such as noun phrases). Text clustering establishes a notion of "distance between documents" and attempts to select enough documents into the cluster so as to minimize the overall pair-wise distance among all pairs of documents. In the process, new clusters are created from documents that may not belong to a cluster.

6.6.3 Bayesian Classifier

Bayesian classifier [11] is a process of identifying concepts using a certain representative documents in a particular category. As an example, one may select a small sample of responsive documents and feed them to a Bayesian classifier. The classifier then has the ability to discern other responsive documents in the larger collection and place them in a category. Typically, a category is represented by a collection of words and their frequency of occurrence within the document. The probability that a document belongs to a category is based on the product of the each word of the document appearing in that category across all documents. Thus, the learning classifier is able to apply words present in a sample category and apply that knowledge to other new documents. In the e-discovery context, such classifier can quickly place documents into confidential, privileged, responsive documents and other well-known categories.

6.6.4 Concept search specification

Effectiveness of concept search in an e-discovery project depends greatly on the type of algorithm used and its implementation. Given multiple different technologies, the EDRM Search specification proposes that a concept search was used for fulfilling a search request and a registered concept-search implementation/algorithm was used, and an identifier (name) of the concept that was used in the search.

Concept search is specified using the operator *concept-search*.

Syntax	Description
concept-search(concept-implementation, x, vendor-param-1, vendor-param-2, ...)	Given a concept x and concept-implementation, locate all documents that belong or describe that concept. Some vendor implementations may require additional parameters.

To indicate the type of concept-implementation, concept search vendors are encouraged to register their implementation name. It is not required to disclose the internal algorithms the vendor utilizes to implement the search. Concept search may be combined with other search constructs, and also be included as part of a phrase or Boolean clause.

6.7 Occurrence Count

Occurrence count search allows a legal professional to specify that a word appear a certain number of times for the document to be selected.

Occurrence count search is specified using the operator *occurs.*

Syntax	Description
occurs(x,n)	For the search word x, count the number of times it appears, and select the document if the specified occurrence count is matched.

6.7.1 Diacritic Specification

For languages that include diacritic characters on certain characters (such as vowels), specifying whether the diacritics should match is a search option.

Syntax	Description
diacritic-sensitive(x)	All the characters must match in their specific diacritic marks.
diacritic-insensitive(x)	Diacritics are not considered in evaluating matches. (This is the default)

6.8 Searching for Parameters

Parameterized search allows searching to be based not on keywords but on certain parameters, such as a document's metadata. Parameterized search is also known as fielded search, because it is frequently performed on data stored within the fields of a database table.

6.8.1 Searching within a Date Range

Date range search allows a legal professional to search a document's metadata to find search results where the creation dates, access dates, or modification dates of documents fall within a specified range of dates. Email usually has an associated To: and From: date, and electronic documents have metadata for the dates they were created, last accessed, and so on. The list of metadata for dates can be found in Appendix 1.

Sometimes there is no information for some of the date fields available for a document, such as when only a creation date exists for no date information for when the document was last modified. Date range searches can be open-ended, where, for example, the search can be for all documents created before or after the date given in the search term.

Date ranges can be specified using the following syntax:

Syntax	Description
= date-string	The exact date is matched

> date-string	The document date is greater than (i.e., after) the date in the date-string
> date-string	The document date is less than (i.e., before) the date in the date-string

Date-string is specified using ISO 8601 standards, which is also adopted by the W3C organization - (http://www.w3.org/TR/NOTE-datetime) in the following way:

YYYY-MM-DDThh:mm:ssTZD

where:

```
     YYYY = four-digit year
     MM   = two-digit month (01=January, etc.)
     DD   = two-digit day of month (01 through 31)
     hh   = two digits of hour (00 through 23) (am/pm NOT allowed)
     mm   = two digits of minute (00 through 59)
     ss   = two digits of second (00 through 59)
     s    = one or more digits representing a decimal fraction of a
second
     TZD  = time zone designator (Z or +hh:mm or -hh:mm)
```

6.8.2 Searching for Metadata

Metadata search allows searching to be constrained based on certain metadata elements of a document. A general search specification allows for naming the metadata fields, specifying the inherent type of that metadata, and the value to search for.

Metadata is specified using the operator *metadata-search*.

Syntax	Description
metadata-search(Name, Type, Start-value, End-value)	Metadata search specifies the name of the metadata using the Name attribute. The Type specifies the type of the values (see section of Type Specification). Also, to specify the matching criteria, the specification includes a Start-Value and an End-Value. A range specification is useful since metadata is typically a numeric valued item.

For a list of named metadata properties see the appendix at the end of this guide.

6.8.3 Searching for Custodian

Custodian search is a common form of constraining search results. To search based on a custodian, the metadata search using the metadata name "Custodian" can be used. Custodian search may rely on assigning custodians to collected data during the Identification Phase so that searching doesn't miss out on custodians. For example, instant messages with buddy-names may be missed if the search term is specified as last-name/first-name or as email addresses.

6.8.4 Searching for Tags

Tag searching allows for specific tags, such as a Batch Label or a Document Tag like "Responsive" or "Privilege", allows the legal professional to filter and constrain results in useful ways. To specify these searches, the following syntax is helpful:

Syntax	Description
tag-search(Tag-Name, Tag-Value)	Tag-Name is any user specified string, and Tag-Value is the current value of the tag.

7. Search Engines and Searching Considerations

Automated searching of ESI requires the specification of a search request in the form of a query. Typically, a search request is issued and the ESI examined to find documents that match the request. Automated search solutions called *search engines* employ different techniques for accomplishing this. While an extensive discussion of these the mechanics of search engines is beyond the scope of this guide, we provide an overview of commonly used techniques.

7.1 Search Engines

Search engines are a category of information retrieval systems designed to supply a subset of items from a population based on a specified set of criteria. The structure and function of search engines are largely determined by:

1) the nature of the information that is the target of a search;
2) the methods used to distinguish a subset of information from a larger population and
3) the interface used to conduct search queries and to display query results.

Most familiar search engines explore the dynamically changing and widely varied information accessible via World Wide Web. In the field of e-discovery, however, the population of information is often static and determined by the scope of electronic data secured during the discovery process. As a result, search engines used in e-discovery can conduct a comprehensive examination of a population of data. Indeed, one of the main requirements of a search engine applied to e-discovery should be the ability to identify all documents and data files that are responsive to a specific set of search criteria.

7.1.1 Search

The defining features of a search engine are the methods by which information about a population of data – metadata -- is collected, stored, and examined to identify a subset of interest. In the scope of e-discovery, the purpose of collecting metadata is to generate a compact and easy to manipulate set of information about a population of electronic data. In addition, the type of metadata collected is intended to increase the efficiency of identifying individual documents based on their unique features or categorizing documents based on common features. To support these goals metadata can take many forms including the type of information contained in individual data files (e.g. spreadsheet, text or images), the relationship between individual data files, information about the origin or source of the data, the dates for the creation and modification of the information within a file or an index of keywords contained in a document. Metadata is stored using a wide variety of methods which can include an index of document contents, a database of document characteristics, or a veridical representation of document contents (i.e. caches files). The functional feature of a search engine is the algorithm used to query the metadata and organize the results of a query. The methods used for searching and representing search results are often proprietary and zealously guarded trade secrets of the organizations that develop and offer search services. Great effort is placed in optimizing the speed, accuracy, comprehensiveness and relevance of electronic searches based on the type of information that is of interest to the user.

The user interface is the most visible aspect of a search engine. User interfaces are responsible for both the collection of information relevant to conducting a search and the representation of search results. The type of information a search engine prompts a user to enter is often a reflection of the metadata and search methods it utilizes. For example, search engines that rely on a full text or content searches of cached and indexed data would be designed to prompt a user for unique key words, keyword combinations/arrangements or strings of text that are found within a population of documents. In turn, the representation the results from this type of search algorithm would place an emphasis on organizing documents according to their unique relevance to the entered search terms. In contrast, a search engine that relies on a Boolean search of a document database would prompt the user for specific search logic that discretely identifies documents. The search terms and search logic are in turn based on metadata or data fields coded for each document in a population of data. The results from this type of search would more likely emphasize a comprehensive listing of the search results since the typical goal is to identify a subset of documents based on their common features that satisfy the specified search requirements. Overall, the most effective user interfaces are characterized by how well they balance the availability of a variety of search methods and search results representations according to the range of searches a user will to conduct.

In summary, search engines aid the process of electronic document discovery by providing a set of tools to both uniquely identify and categorize documents. While the overall capability of search engines are constrained by the distinguishing characteristics of electronic data, a wide variety of data collection, data organization and data query methods are employed to assist users in identifying documents of interest.

7.1.2 Fielded Search (Searching Metadata)

Fielded searches are based on values stored as metadata rather than actual content of an electronic asset. Searches can be refined using metadata information extracted during processing, such as sender or receiver, creation date, modified date, author, file type and title, as well as subjective user-defined values that may be ascribed to a document as part of downstream review. Examples of subjective field values could include designations for relevance, confidentiality, and custodial ownership. Coded values can be used to both add distinguishing information to a document or to code document features that can be used to categorize a set of documents. In this way, field values can be used individually or in combination to create highly specific results sets.

The effectiveness of a fielded search is predicated on the degree to which the query has been refined. Most fielded searches can be refined by using operators to join combinations of fields, as well as operators that expand or limit queried field values. For example, Boolean operators can be used to influence the inclusivity or exclusivity of a search query. A query using an *OR* operator between a specific file type and a specific author (e.g. *Microsoft Excel OR Author A*) will generate a far more inclusive set of results than a query for (*Microsoft Excel AND Author A*). The use of 'OR' would, however, be highly effective when searching for a single file type from a number of authors - *(Author A OR Author B OR Author C) AND Microsoft Excel*.

Whereas Boolean operators can be used to join fields, other operators such as EQUALS, BETWEEN, CONTAINS and LIKE are used to refine field value searching. For example, a search using *EQUALS* will return only exact matches for a specific field value – 'Title' EQUALS 'Second Quarter Sales Report'. A reviewer may use the operators LIKE or CONTAINS to construct a more inclusive query for a broader set of field values. The operator LIKE runs what is termed as a '*fuzzy search'* which returns approximate matches of the queried field value – Title LIKE '2nd Quarter Sales Report'. 'LIKE' is useful because it will allow the retention of search terms despite any spelling errors or other variations in the naming scheme found in a document. 'CONTAINS' will return any file where the specified keyword appears as part of the field value. The search 'Title' CONTAINS 'Report' will return 'Second Quarter Sales Report' in addition to any other document with the word 'Report' in the title. This approach can be a very useful approach when the specific name of a document is unknown.

7.1.3 SQL and Other Query Languages

7.1.3.1 What is SQL?

SQL is an acronym for Structured Query Language. In the scope of e-discovery, SQL provides a syntactical framework that is used to input, manage and query the information stored in a relational database management system (RDBMS), typically referred to as 'structured content' as opposed to unstructured or semi-structured content. To gain an understanding of what SQL is and how it is used by some document management systems, it is helpful to first have an insight into the nature of the relational databases it operates upon.

Relational databases provide a logical structure that both stores varying types of information (i.e. data) and represents the relationships between distinct pieces of information. In SQL, data is classified according to three levels: Class ➔ Category ➔ Data Type. Some examples of classes include data representing numeric values, ASCII or Unicode characters, or chronologic values. Categories under each of these classes separate data according to operational features. Some data features include varying or static lengths for strings of characters, decimal or integer values for numeric data and date or time expressions for chronologic data. Finally, data types provide a finer level of granularity that specifies the exact format that data is stored in the database.

Relationships in a relational database are represented by linkages that exist between two or more pieces of data. For example, in the table below each row represents the distinct features associated to a single person and each column represents features that are coded for each individual. In this way, data that is unique to an individual such as age, name and birth date will always be linked in the database. In turn, each entity – in this case a person – that is entered into this table is linked to a set of characteristics specified by the structure of the database.

ID	First Name	Last Name	Age	Birth date	Blood Type ID
1	Jim	Smith	5	2003-12-01	2
2	Cindy	Walker	21	1987-12-28	2
3	Mark	Bush	72	1936-08-19	8
4	Peter	Hamden	34	1974-03-26	5
5	Rebecca	Larson	31	1977-04-27	6

The first defining characteristic of SQL is illustrated by how the language is used to create a relational structure among data types and enter specific values for each entity represented in the table. For example, the following statement is used by a SQL based RDBMS to create the table illustrated above:

```
/*
 *  Create a table to hold various patient data
 */

CREATE TABLE [dbo].[Patient_Data](
      [Key] [bigint] NULL,
      [First Name] [varchar](50) NULL,
      [Last Name] [varchar](50) NULL,
      [Age] [smallint] NULL,
      [Birth date] [datetime] NULL,
      [Blood Type Id] [bigint] NULL
```

Once the structure is created, SQL provides syntax for populating tables with data:

```
/*
 *Insert a record into the Patient_Data table
 */

insert into dbo.Patient_Data
values
(1,'Jim','Smith',5,'2003/12/01',2)
```

The second defining feature of SQL is its ability to manipulate data within a database. For example, the following statement can be used to increase the age by one year for all the people listed in the table.

```
/*
 * Increase all ages by one year
 */

update Patient_Data
set age = age + 1
```

The final defining feature of SQL is its ability to return data from one data field based on its relationship with another data field. For example, the following query will return the first and last name of individuals listed in the table above based on their age.

```
/*
 * Retrieve First Name and Last Name of all
 * individuals aged five years or older.
 */

select [First Name], [Last Name]
from Patient_Data
where age >= 5
```

In the field of e-discovery, RBDMS are used because of their ability to store large amounts of data in a compact format without losing any information. For example, in the table above, blood type is represented by a single ID number. The following SQL query can be used in combination with the table below to retrieve each person's blood type.

Blood Type ID	**Blood Type**
1	O Negative
2	O Positive
3	A Negative
4	A Positive
5	B Negative
6	B Positive
7	AB Negative
8	AB Positive

```
/*
 * Retrieve First Name, Last Name and blood type for all
 * individuals in the database
 *
 * Note: A and B are table aliases which are used to
 *  save retyping table names.
 */

select [First Name], [Last Name], [Blood Type]
from Patient_Data "A" inner join Blood_Types "B"
on A.[Blood Type ID] = B.[Blood Type ID]
```

In this example the single numeric value "8" can represent a string of 11 characters "AB NEGATIVE". While the savings in data storage may not be apparent in this example, it is possible to see when considering a table listing information for millions of people. Specifically, if an integer only takes up 4 bytes of storage space vs. 20 bytes for a character string the potential storage savings over 1 million entries, for a single column in the table, is 16 megabytes. Savings in storage space also often translate to faster completion of search queries. Furthermore, minimizing the amount of redundancy has the added benefit of simplifying data management. Continuing with the example, a typographical error during the process of entering blood types would only need to be fixed in one location when data is split across two tables. This is clearly less time-consuming than attempting to fix 1 million patient records where the blood type is stored along with personal information.

To perform searches on large databases with adequate performance it is important to create the proper indexes. This includes columns where JOINS are typically performed. In addition to the relational data, databases support indexing of data based on data type and are often efficient at sorting information that is indexed, e.g. a date column. Some databases also have full text indexing capabilities built-in.

The true power of SQL becomes evident when important facts and trends are hidden among massive amounts of data. The combination of SQL and a powerful RDBMS platform allow data to be represented, manipulated, classified, and summarized in a standard and robust manner. SQL can also be used to ingest different datasets to determine overlap or disjointedness. SQL provides a rich and diverse set of tools for handling many data-related tasks.

7.1.4 Indexing

Indexing is a process that inventories the total content of a file. The end result is similar to the index at the end of a text book. Without an index, the process of searching for a specific word or phrase would involve a page-by-page review of the text for each new query. An index allows a reader to quickly and efficiently locate pages containing a specific term or phrase. Search indexes serve precisely the same function as tools designed to facilitate and expedite the retrieval of information.

As data is indexed, the content is scanned to identify unique terms and term locations within the text, and perform additional functions against the data, such as stemming algorithms, ranking, natural language or conceptual modeling. This information is then retained in a database or structured search catalogue that is specific to the search engine used, and can be queried using any standard structured query language. Querying an index can involve a simple search for a single keyword, or a more complex query involving multiple keywords and proximity restrictions. The syntax and format for queries depend on the conventions used by a search engine or the database language used to query the indexed data.

Search engines will use both common and proprietary technology to build indexes and service search queries,

For the most part, it is not practical to index every term in a file. Certain words are so commonplace as to offer little or no value when conducting a content search. To avoid creating an overly inclusive index, most indices utilize a *noise word* filter. This filter includes a customized list of terms that are overlooked during indexing. Each index has a list of ignored noise words that can be customized. Some common noise words include ‘a’, ‘and’, ‘the’, ‘from’, and ‘because’.

7.1.4.1 Term Selection

Term selection determines what terms in a document are indexed and findable through the search engine. Some search engines use complete term indexing covering all terms while others use partial term indexing, with differing techniques to eliminate some terms before indexing by the search engine. Terms that are eliminated before indexing cannot be searched on so care must be used when eliminating terms to index. Complete and partial term indexes have different advantages and disadvantages which are described below.

Organizations procuring search technologies should ask what type of term selection is used. If noise words (black lists) or pre-selected words (white lists) are used, the lists should be available to the users and may need to be delivered to the opposing party for increased transparency.

Term Selection	**Description**
Complete Term Indexing (index of all terms)	**All Terms Can Be Searched** Complete term indexes cover all source terms and are the most comprehensive method of term selection. They ensure that every term in the files can be successfully searched, eliminating the problem of false

	negatives and additional noise which can occur when words are removed from the indexing process. **Accurate Search** Indexing all terms provides the most accurate search and reduces both false negatives and false positives that can exist with other approaches. False negatives are reduced or eliminated by indexing all terms because searching on any term will find documents that have that term in them. No terms are removed to prevent accurate search. False positives are reduced because term elimination also removes terms from search queries. For example, searching on the phrase “vitamin a” where the word “a” was eliminated from the index and query would return all documents with the word “vitamin” no matter what vitamin was being discussed.
Partial Term Indexing Using Noise or Stop Words (black list)	**Noise or Stop Words** Noise words, also known as stop words, are typically common words that some search engines use as a black list for term removal when creating the search index. The reasoning behind this is that some search engines consider some terms to have little or no value and choose not to index them, hence the name noise words. Some common noise words include ‘a’, ‘and’, ‘the’, ‘from’, and ‘because.’ Noise words vary by language so search engines using noise words must correctly identify the language being indexed and select the appropriate black list. Products that use noise words often make the black lists available to their users so one can be informed of what words are not indexed and cannot be searched. **Reduced Information (False Negatives)** A significant issue with noise words occurs when you want to find one or more words that have been removed from indexing and thus cannot perform the

	search you want. A well-known example of this is the phrase "to be or not to be." By themselves, each word in often listed in a noise or stop word list but, together, they are obviously meaningful. Traditional use of noise words could render this phrase unfindable using the search engine. Other examples where meaningful information may be eliminated include "vitamin a," "stock symbol k," "C++," etc. **Increased Noise (False Positives)** When noise words are used, they are eliminated from not only the search index but also the search query. Automatically eliminating words from the query can return documents that one was not expecting to receive, producing false positive results. For example, searching for "vitamin a" may have the "a" removed, returning all documents with the word "vitamin."
Partial Term Indexing Using Pre-Approved Words (white list)	**Pre-Approved Words** The use of pre-approved words uses a pre-determined list of all words that will be indexed. This list functions as a term white list. When white lists are used, words that are not on the list are typically not indexed and cannot be searched on. While this type of indexing typically provides the least search capability, it is used by some popular products so it is important to understand the characteristics of this approach. It generally provides the smallest index; however, it achieves that by eliminating all words from the index that are not on its list. **Reduced Information (False Negatives)** Partial term indexes using pre-approved words can dramatically reduce the amount of information indexed, much more than the use of noise words. While noise words remove information that is often, but not always, of little use, pre-approved white lists may miss many important words.

	For example, if a search index was created without pre-approving the name 'Katrina' or the word 'Hurricane' searches for 'Katrina' and 'Hurricane Katrina' would not return any results. These words will often not be missed using noise words, but it is likely they would be eliminated using pre-approved white lists. **Increased Noise (False Positives)** Pre-approved white lists can result in more false positives and documents for review than either complete term indexes or nois word approaches.

7.1.4.2 Additional Indexing Customization

Most indices can be customized to better meet specific searching needs. Custom setup preferences can include the sensitivity of upper- and lower-case letters, recognition of specific date formats, the inclusion or exclusion of specific file types, and Unicode compliance to identify and index foreign characters. Each of these customized parameters will influence the manner in which data can be searched, but may also allow for more comprehensive indexing.

Similar to noise word filtering, nearly all punctuation in a document is ignored during indexing. Characters such as periods, quotations, ampersands, tildes, commas and brackets are all indexed as empty spaces. Depending on the search term syntax used by a search engine, specific punctuation may be recognized as a search operator. The inclusion of punctuation as a search operator, however, does not impact the way in which a document is indexed. Derivations of a root term can also be queried by adding other characters such as asterisks, exclamation points and percentage signs. For example, a search for *legal** would generate hits for *legal*, as well as *legality*, *legalities*, *legalize*, etc. Another example would be the addition of a tilde to the end of a word to search for all synonyms of that word.

7.1.5 Streaming Text

The streaming text method of searching does not pre-index the content or metadata, but instead reads the text of a document one keyword at a time, from start to finish Each term is examined, and matched against a the query in real time, if there is a match, the document is considered to be a search hit.

This technique closely mirrors how a human would look for keywords in a printed document. But since automated search using this technique is likely to be limiting, certain variations on the scheme are available. Some known variations on this scheme are:

a) Place a wildcard character in the keyword of the search query in order to improve the search results by including wildcard matches.
b) Provide *regular expressions* matching to expand the search.

However, this technique should be used with care. Typical challenges with this technique are:

a) The entire content needs to be completely scanned, for each search, causing the search to take very long time.
b) The process (full scan) is repeated for each consecutive search
c) This has the potential to alter the metadata of ESI.
d) Many ESI types are not searchable, since the data is stored in a form that can not be streamed and searched. As an example, Word documents and Excel spreadsheets are not easily streamed for searching of keywords.
e) Expressing complex searches or results is difficult.

7.1.6 Data Type Specifications

Electronic information will nearly always exist as a combination of structured data elements and unstructured content. “Structured” content is any overtly labeled element; metadata, or field such as the sender of an email, creation date, or tag applied to a document during review. Databases are designed to manage, organize and automate processes based on these structured content values. Unstructured content, such as the body of an email, or audio file is typically where the majority of relevant information resides, it is not tagged or parsed, so search technologies are used to query the content or an index of the information it contains, and present the results.

Servicing e-discovery will commonly combine fielded (structured) search with the ability to query the unstructured content. Certain fielded search specifications require using a particular data type. Unlike keyword searches on unstructured content where a keyword is a simple string, data types for identifying and applying fielded constraints may require specifying data types. As an example, search for a metadata such as Last-Modified-Time of a document, the search criteria needs to specify values in the form of DATE.

Table of Data Types:

Data Type	Description
INTEGER	Integer Data Type specifies that the value is a simple number.
FLOAT	Float Data Type is used for specifying fractional numbers, using IEEE 754 Floating Point Number definition.
DATE	Date Data Type defines the date values, using ISO 8601 Date Format specifications.
STRING	String Data Type allows specifying string parameters for

	search, as in keyword searches, but associated with the metadata.

7.2 Language and Content

As companies increasingly become multi-national, litigation involving non-English electronic documents and e-mail becomes more and more common as well. Computers are able to store each of the different alphabets and character sets for all of the existing languages, but in order to use these non-English languages some technical considerations must be taken into account that can affect how search is performed and what the results are.

7.2.1 Character Encoding Specification

All electronic data is represented as sequences of bits, or numbers. Therefore, each alphabet or script used in a language is mapped to a unique numeric value, or ‘encoded’ for use on a computer using a standard known as *Unicode*. Each letter or character has been assigned its own unique value in the Unicode encoding schemes, known as the Unicode Transformation Format (UTF). The UTF utilizes multiple encoding schemes, of which the most commonly used are known as UTF-8 and UTF-16. For example, the English alphabet and the more common punctuation marks have been assigned values between 0 and 255, while Tibetan characters have been assigned the values between 3,840 (written as x0F00) and 4,095 (written as x0FFF). All modern (and many historical) scripts are supported by the Unicode Standard. Unicode provides a unique number for every character, regardless of the platform, program, or language. The Unicode Standard is described in detail at the website http://www.unicode.org.

When deciding to store and search non-English documents, the following points need to be considered:

a) The search system needs to be able to support Unicode, since some systems were created to support text encoding schemes which predated Unicode.
b) Some of the more common non-English languages, particularly Asian languages such as Chinese and Japanese, require two bytes instead of one byte in order to store a single character. A multi-byte encoded document could require twice the storage space of a single-byte encoded document with the same number of characters. This is an important consideration when allocating storage space for multi-byte encoded documents.

7.3 Specifying Case

In general, keyword searches match words in documents without considering whether any or all of the letters in the keyword or the documents are uppercase or lowercase. If these are important for search, the search specification must include them.

Specifying that the search must be case sensitive will match the exact case for all letters in the keyword and in the documents. For example, a case-sensitive search on *AIDS* will match the word *AIDS* in the phrase “increased number of cases of AIDS” but won’t match the word *aids* in the phrase “the nurse aids the operating room surgeon”. Similarly, a case-sensitive search on *Rose* will match the name “Rose Jones” but won’t match the phrase “rose garden”.

7.3.1 Language Specification

When using a collection of multi-language documents, the collection may contain not only documents written in several different languages, but also single documents that are themselves written in two or more languages. In either case it becomes necessary to specify which language the search terms belong to.

For example, if the search term entered is *pan*, this can mean “bread” in Spanish as well as “pan” in English. Similarly, *son* can mean “its” in French and “son” in English. Specifying which language the search term is intended to belong to will affect the search results. In a similar vein, the differences between British English and American English can affect the result set if the wrong term is chosen, such as using the American term “trunk” instead of the British term “boot”.

In addition, search engines may have noise words for each supported language. Just as some search engines eliminate high frequency English words, such as *a*, *and*, *the*, a search term that may be meaningful in one language may appear as a noise word in another. Again, specifying the language of the search terms will affect the search results and it is important to get the list of noise words by language used by the search engine being used.

7.3.2 Tokenization Specification

Before a search can occur, a search engine needs to take the text in a document and break the text into searchable keywords. This process is known as *tokenization*. Tokenization involves identifying special characters such as blank spaces, commas, or periods and using them as separators between words. As an example, if a document contains “cats or dogs”, the tokenization looks for spaces and creates three keywords: “cats”, “or” and “dogs”.

7.3.2.1 Special Cases with Tokenization

1. The hyphen (-) character. The following list contains legitimate words:

e-mail
editor-in-chief

well-respected

Each of these is considered a word, but also contain within them distinct words. Search engines may tokenize each of these as one word, multiple words, or both, depending on the algorithm the search engine uses.

2. Chinese, Japanese, and Korean (collectively CJK) languages. While most languages use blank spaces or other special characters to indicate breaks between words, a few do not. CJK languages, for example, are written with all characters side by side, and the words can be made up of one, two, or more characters. In addition words can be made up of other words. Context determines how to break the characters into words.

There are two major methods to tokenize CJK languages, N-gram and dictionary-based approaches, each with their own characteristics.

The N-gram method breaks sentences into individual characters regardless of words, and matches each character in the search string with the character in the document. This allows the search engine to index and find all words in the languages with minimal overhead. In this approach, the Japanese word for "system" エンジン would be indexed as エン, ンジ, ジン, ン. By indexing all characters without the need for a-priori knowledge of the language and words, the search engine can guarantee that all terms in the file can be searched on. A consequence of guaranteeing that all words can be found is that the approach can result in a larger search result set than dictionary-based approaches which may not be able to find all words.

The dictionary-based approach uses lists of known words to tokenize CJK languages into proper words. The use of dictionaries can work well to reduce search result with common words when the risk of not finding a document (false negatives) is low. However, dictionaries are only as effective as their word coverage. Dictionaries are maintained by a few organizations and often run in the millions of characters; however, it is acknowledged that this is not enough to cover all words as new general words and proper nouns are constantly being created in these languages. Additionally, dictionaries need to cover many uncommon word variants which exist in CJK languages. For example, the word island which can be represented in Japanese as 嶋 or 嶌 instead of the usual 島. Although it is impossible to have a fully up to date dictionary, dictionary maintainers periodically release updated versions. When a new release of a dictionary is available, it is necessary to re-index the corpus to find the new added words.

In general, the N-gram approach can be relied upon to produce complete search results while providing a larger result set while dictionary-based approaches can reduce the result set so fewer documents need to be reviewed, but by increasing the risk that some files may not be findable. The advantages and limitations of each approach should be understood by the producing and receiving parties.

3. German. Some languages, such as German, are made up of compound words that are not hyphenated to indicate the word breaks. In order to tokenize these languages, the search engines must decompound the long words, breaking them into their components.

4. Diacritical marks. Some search engines conflate similar characters that differ only by diacritical marks into the one letter, which can cause expanded search results. For example,

7.4 Document Segments and Segment Scope Specification

Documents often contain a certain structure to them, and it may be important to consider document segments and restrict the search specification to a scope. There are a number of considerations here.

a) Scope is for a particular keyword or phrase
b) Scope is for the entire Boolean expression

Keyword or phrase specific scope is indicated using a prefix in front of the keyword as per the syntax:

Syntax	Description
segment: keyword	Keyword should be present only in the specified segment.
segment: phrase	Phrase should be present only in the specified segment.
(segment: (Boolean-expression))	The entire Boolean expression must match only within the segment for the document to be selected.

E.g., to search for a multiple Boolean in two segments Title and Abstract, a search string of the form: (Title: (report and pets)) AND (Abstract: (pet-owners AND (cats and dogs)))

8. Documenting Search Results

An important aspect of search is the documenting of search results for each search. Documenting search results enables several important follow-on actions as listed below.

a) Defensibility of search results (see next section for details on various methods of validating systems and actual search process).
b) Communicating search methods and results both within internal legal e-discovery teams and to outside parties such as outside counsel and opposing parties.
c) Monitoring and historical tracking progress of searches.
d) Assessment of search strategies, search technologies and specific vendor selections.

The EDRM Search XML Specification presents a formal way of documenting the search process and authenticating search results. This section describes the specific items that are recommendations on various items that should be captured.

8.1 Results Overview

In order to capture the essential elements of search results, the following broad areas of search process and results need to be considered for appropriate documentation. As every matter differs in scale and expected level of reasonable diligence, the e-discovery team should review these potential metrics and documented actions.

- Overall document counts in the comprehensive ESI corpus that was searched.
- Collection substrata ESI Document counts broken by file type, custodian, date ranges.
- Loading state of the target corpus that indicates the batches or collections loaded, processed or staged.
- List of search queries, with a complete query specification that captures the search technology, search parameters, search user, the time of search.
- For each search query, capture meta-data of search results
- For each search result, identify the hit location within the result document – i.e., hit level results specifying where within the document a search term was found. Some systems highlight these terms for easy viewing and navigation.
- Additional overall search aspects such as the language of documents within the corpus, the regions in the document that were searched and the regions that were not searched and the regions that could not be searched
- Search accuracy metrics for each search and the overall matter.
- Search performance metrics in terms of items searched per unit of time and items retrieved per unit of time.

The following sections expand on various aspects of the above items. The e-discovery team should evaluate and adopt these potential documentation steps in light of the matter venue, scope and standard of care required to authenticate your search results.

8.1.1 Overall corpus counts

When recording search results, it is important also note the actual size of the corpus searched. This gives a perspective on selectivity of a search query. In general, overall size of the corpus is best indicated by the number of documents present in the corpus.

In order to meet goals for validation of search results (as outlined in Section 9), it is important to record and report the overall corpus size. As an example, if a search were to process 100 million documents and yields 10,000 documents as “hits”, the search can be considered quite effective in culling down the results. In contrast, a search that processes 20,000 documents and yields 10,000 documents may not be as effective.

If possible, the total individual or unique item count within the target corpus should be compared to the items reported during collection or processing. Technologies may interpret container items differently (examples include sub-attachments and embedded files within email). This can make such comparisons difficult, but the goal is to document the actual number of items searched and demonstrate that this count reflects all the items in the corpus minus any documented exceptions.

Additionally, the documentation of search results should also include document counts that are compound complete, or including all family members. For example, one attachment may contain a search “hit” but if the email family includes a parent email with 14 attachments including attached emails that also have attachments, then the search result including full families would be much larger than the one recorded “hit.” It is quite helpful for the e-discovery team to know the total number of documents returned including all parents and attachments as all of the resulting items might then need to be reviewed.

8.1.2 Corpus breakdown

A breakdown of corpus into various categories is useful in analyzing and validating a search technique. Primary value is in categorizing an input corpus so that search hits within individual categories give the legal team the tools to further iterate and refine a search methodology. Often, this breakdown is referred to as corpus stratification and categories are called strata. Some of the useful breakdowns of corpus are listed below.

a) By custodian
b) By date range
c) By tags (such as manual coding)
d) By file/document type
e) By loading state (see below)
f) By file meta-data (such as document author, document title, document size)
g) By language of documents
h) By de-duplication status
i) By previous search results

For each category above, the search hits in each category will be helpful in further analysis.

As an example, a search for a keyword may locate search hits predominantly in a certain date range. Also, in the date range where the search hits are located, the corpus may itself have a certain number of documents. Together, this information is useful in either narrowing or eliminating a certain date range from further consideration.

Another interesting breakdown is how current search results are distributed within a previous search results. If you have a broad search result that identified potentially relevant results, a new search that identifies more specific results within that search result is useful. Also, if that same search result identifies a different number of search results outside the previous search hit coverage, that fact can be used for further validation.

8.1.3 Loading states and batches

Corpus and search results breakdown by loading state is a useful technique. Quite often, ESI for a legal case is brought into the case in batches or tiers. These batches reflect multiple collection scope or methodologies along with importance of certain ESI relative to other ESI. By breaking down the corpus of documents by batches as well as identifying search hits for a search across these loading batches, the legal case team can establish useful conclusions that can further drive their additional searches.

Because searches will be executed on incomplete batches or ESI collections, it is critical to record the status of the target collection so that supplemental searches can be run on subsequent batches or that target corpus can be searched again if additional criteria are added.

8.1.4 Search query recording

As part of search results recording, the search query that generated the results need to be recorded. The search query needs to be recorded in a form that allows re-execution of the search. A primary purpose of this is to allow testing and validation of repeatability of a search along with a correlation of the search query against the results.

Query recording needs to include all parameters of the search, including the search technology, the corpus that is targeted, specific meta-data properties, document regions, language, stemming, tokenization, and other properties.

Many search systems record this information, but the e-discovery team should confirm that every aspect of the query is logged and not changed when system defaults or parameters are subsequently modified.

8.1.5 Overall search results meta-data

Search results meta-data breaks down the search results in a form that allows a quick review of the results. Some of the items captured are:

a) Total search hits in the form of documents which were identified. In some cases, document hits may be available on unique documents (i.e., after de-duplication).
b) Total number of keyword terms hit in each document (if the search technology is keyword based).
c) Total number of documents in each corpus stratum.
d) The time when search was initiated.
e) The operator that performed the search.
f) The duration the search ran/executed.
g) Document counts for exceptions (those that the search could not process).

8.1.6 Document-level search results

Document-level search results enable the legal case team to track down a specific document where a search hit occurred. Also, a complete document-by-document review of search results requires identification and eventual retrieval of the document in a reviewable form, again necessitating such results.

On a practical level, a document level results report is essential for later authentication of potential evidence. Consider it part of the overall Chain-of-Custody that documents how these documents were selected and where they came from.

Document-level search results should also include an integrity check based on a hash computation of the document. Typical search systems perform this integrity check using an MD5 or SHA1 message digest of the document and storing that hash value. When document hits are identified, the document ID along with the hash value and its location (i.e., a pointer to where the original document exists) is captured. The EDRM XML standard defines several other properties for each document using XML Elements.

In addition, document-level search results may include one or more small portions of the document (snippets) to indicate the context of potential hits within a document. These snippets enable a review team to perform a quick review of search results.

8.1.7 Hit-level results

When a search identifies a document, the search operation has scored a hit. In the case of a search query that is based on a search term, it is possible to have that term be found multiple times within the same document. It is also possible for a search query that contains multiple keywords to have one or more search hits for some number of terms. If these keywords are connected by Boolean operations, it is possible for some subset of keywords to be present, but it may not constitute a hit since the entire query may not match the document contents.

Search results at the hit level therefore capture the keyword, and its potential hit position within the document. For queries that are not based on keywords (such as concept search), it may not be possible to identify a hit. However, if there is a sentence or paragraph level context that was responsible for the document to be selected, that is captured using a hit. The EDRM Search XML Guide captures hits in the form of Hit Position Descriptions. These report the locations within various documents where a particular hit was found. It provides positive confirmation that a particular search actually exists, without revealing the complete contents of the document. Remember that any subsequent conversion or reformatting of ESI may affect or negate any positional descriptions.

In addition to the document itself, the section within a document where a search hit occurs is also important to capture. Examples of keyword hit locations are:

- Body
- Header
- Footer
- Comments
- Track Changes
- Hidden Cells/Columns
- Formulas/Links

- Document internal metadata fields
- File System meta-data fields
- Multilayer text – such as text below a PDF/TIFF image
- Image tags and other meta-data of other objects

In the case of fielded searches, the fields where a search query was applied, the actual field values need to be recorded as well. Additionally, some searches may produce a hit within a container of other objects (such as an email and its attachments). In these cases, search results should capture both the container object references as well as the contained object references so that a document hit can be correlated with the actual document/container.

8.1.8 Keyword occurrence counts

Another useful aspect of search results is recording of counts of keywords that appear within the entire search results. When a search query involves multiple keywords or when one or more of the queries produces stemming, wildcard or fuzzy-based variations, a complete count of total occurrences for each keyword is useful for evaluating the value of searching using certain keywords. In some instances, the keyword counts both at an aggregate level (totaled over all the variations) as well as counts based on an individual variation level would each be helpful. If a search query is based on other related terms, the search results should capture the occurrence counts for each related term. If exclusion criteria are used to filter out known non-relevant documents, these occurrence metrics can provide a useful way to monitor for changes in language usage over time or sources.

8.1.9 Language, Script and Encoding

Managing search results in a multi-language corpus can be tricky. A useful operation is to isolate the search results by language allowing specific language experts to review these results. To facilitate this as well as to allow for easy query iterations, search results should be categorized along languages. Additionally, some search systems may be able to categorize the language script used as well as how the characters of the language were encoded. Language encoding is a standard way to represent characters of a text document, with Unicode Encoding being one such standard.

8.1.10 Search accuracy measures

Search results accuracy measures are an indication of how well a particular search performs. The standard measures are Precision and Recall (described in Section 9). In many instances, precision and recall are difficult to compute before a document-by-document review is completed.

A useful method for overcoming this is to use a known corpus and evaluate the accuracy against the expected results for this corpus. An example of this would be to take proposed criteria for possible privileged documents and executing the search on a prior corpus that has already been reviewed. An alternative is to create a smaller sub-corpus through random or manual selection and use this after performing a manual review.

8.1.11 Search performance measures

Search performance measures are based on how long each search takes to complete. The EDRM Metrics Project documents Search Performance measures under the Analysis Node. Typical measurements are in the form of number of milliseconds to complete a search query.

There are no standard expected search performance measures, since each search method could involve varying levels of complexity, and the time taken to complete a search would be a function of this complexity. It is still useful to record the actual search query, the overall corpus size/counts, the number of actual search hits found, and the amount of time it took to complete.

9. Validation of Results

Validation of results is an important phase of search. Some of the overall goals of this phase are the following.

a) Ensure in a cost-effective way whether a set of searches performed are satisfying a production request. When producing very broad searches, it is often difficult to perform a large-scale human review, so the validation phase should provide the necessary evaluation without consuming too many review resources.
b) Ensure that the validation produces enough results in a timely way, to assess and evaluate whether we need to modify the initial set of searches (i.e., assist in the feedback loop to the Execute phase of search).
c) Allow for comparison of alternative search methodologies.
d) Support the needs of EDRM Metrics in terms of tracking and feeding processing and analysis metrics.

Measures for validating results depend on the overall goals of the e-discovery production. In particular, the following considerations apply.

a) When searching for responsiveness review, validation of results may need to consider whether the overall goal is to be restrictive or over-inclusive. Depending on the case situation, this may drive the evaluation. While certain documents would have a clear-cut responsive determination, some would not. Over-inclusive strategy would include those that are not clear-cut responsive. A restrictive strategy would discard these. In some cases, the cost of human review and overall budget may impact this choice.
b) When searching for privilege review, a strict validation step may be required, causing a feedback that makes the searches more broad in selecting results. However, this may be subject to consideration when claw-back agreements, selective waivers, and FRS Rule 502 agreements are in place – i.e., the privilege search may be stricter.
c) When evaluating whether it is necessary to expand the ESI collection (perhaps adding new custodians), validation of searches may need to evaluate the search

results in the context each collection batch. Quite often, the tiered collection of ESI results in some batches to produce larger number of responsive document hits compared to other batches. Thus, it is necessary to properly document and compare search results.

9.1 Search Results Descriptions

To validate search results, the following needs to be captured:

a) The search query that was submitted.
b) The number of documents that were found.
c) The number of documents that were found to be duplicates of other documents.
d) If a document is contained in another document (Emails and attachments etc.)
e) The number of documents that were searched.
f) An identification of each document (using MD5 or SHA1 hash content).
g) The number of hits within each document.

The above results may need to be classified across additional lines such as below.

a) Custodians where the results were found.
b) Loading batches where the results were found, so that search results are tracked per-batch.

9.2 Validation Methodologies

There are several validation methodologies that should be used throughout the development of the search criteria to be used for selection of documents for attorney review. Many of the below-referenced validation methodologies involve the case team in reviewing samples of documents to determine litigation relevance to classify documents as Responsive or Not Responsive to the issues of the case and therefore increasing the precision of the search results.

9.2.1 Frequency Analysis

Initially, frequency analysis may be used to evaluate the effectiveness of the initial search criteria. The search terms are tested to determine whether they effectively discriminate between potentially relevant and clearly non-relevant data. Think of this as a reality check on the search results versus the overall collection size and the reasonably expected proportion of relevant results. If the collection consists of ESI manually designated as relevant by custodians, an 80% response rate might be reasonable. Whereupon if you are searching across the combined departmental mailboxes and file shares, you would expect a much smaller result set.

This method involves reviewing the proportionate counts of items returned using the initial search criteria set and individual search terms. Depending upon your search technology, it is useful to be able to evaluate the overlapping search terms and see which items only received one or multiple hits.

Once you have identified overly broad terms, samples are used to develop valid qualifiers or exclusion terms that may be used in combination to focus or narrow the search. Non-discriminating criteria, those terms that are over-inclusive and are not likely to yield responsive documents may be removed.

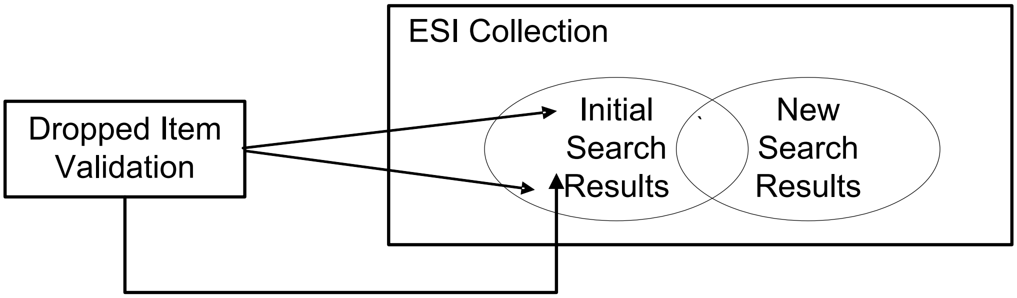
This analysis also identifies search criteria that fail to retrieve any data or fail to retrieve the quantities or types of data expected to determine whether these may need broadening or further investigation.

This process is used iteratively throughout the life cycle of a project as search criteria are modified. The goal of this method is the increase the relative precision or proportion of relevant items within the search results. It does not address the recall or completeness of relevant items out of the collection.

9.2.2 Dropped Item Validation

As the search criteria set is being updated and modified during the initial investigation and analysis, dropped item analysis is another form of validation needed to ensure that Responsive items are not being inadvertently omitted through changes to the search criteria.

This comparison would sample documents that were originally results of one search criteria set but are no longer results of the modified search criteria set.



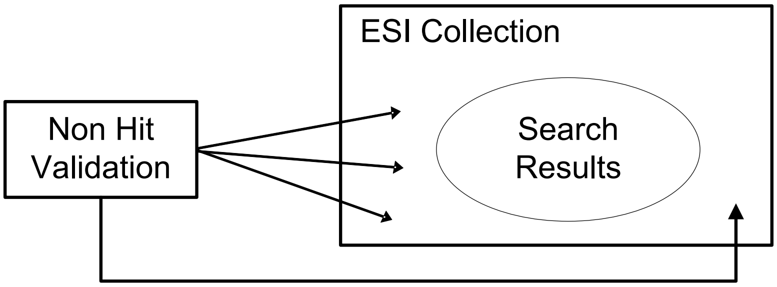
The case team would then review the samples of dropped items for responsiveness to ensure that Responsive items had not been dropped. If Responsive items are identified, they should be reviewed to determine whether additional terms need to be created to capture these items or if modifications made to the criteria should be changed so these items would still be included.

The appropriate number of random or statistical (example: every 20th item) items sampled is discussed in Section 8.5. Sampling checks should be repeated after the criteria have been modified until the team is satisfied that the threshold of confidence has been reached. Dropped item validation can be performed in combination with the following

‘Non Hit Validation’ sampling method, but items that fell within previously responsive search criteria should have a higher threshold of acceptable error.

9.2.3 Non Hit Validation

As search criteria are being developed and used to move resulting items forward into document review, samples should be taken of items that did not hit on any of the search criteria being used. The case team should review the non hit items for responsiveness. As with the Dropped Item Validation, any documents that are deemed Responsive during review will need to be evaluated to identify additional search criteria or modifications needed to existing criteria to capture these items to move them forward into review.



As with the Dropped Item Validation, any sampled documents that are deemed Responsive during validation review will need to be evaluated to identify additional search criteria or modifications needed to existing criteria to capture these items to move them forward into review. These methods of sampling the items outside of your results are critical to building a defensible process. They demonstrate reasonable efforts to ensure that everything responsive has been found. In comparison to the Frequency analysis, these methods improve the search recall (proportion of relevant items versus all relevant items). Using these methods decreases the risk of inadvertently missing relevant items, but will increase the total volume for review.

9.2.4 Review Feedback Validation

Documents in the review set are reviewed by attorneys for responsiveness, privilege, and other issues involved in the specific matter. Feedback from the review – i.e., calls by reviewers about which documents are relevant or privileged – can provide additional information useful in refining the search and selection criteria or in identifying gaps that require additional analysis. This feedback will be used for additional analysis and to refine the Search Criteria sets. The feedback may identify categories of documents that are not yielding responsive documents. This information will be used to develop exclusionary criteria that will identify documents to be excluded from the review set. Also, the feedback may identify new categories of documents that should be included and the criteria will be broadened to include those documents in the review set.

An example of Feedback Validation would be to do a complete review of a key custodians collected ESI and then compare the relevant or privileged documents against the search criteria frequency metrics. So if a particular search term appears in 80% of your relevant documents and not in any of the non-relevant documents, that term should be used to focus your narrow set of known relevant criteria for first pass review. Depending upon the reporting capabilities of the search system, it may be necessary to segregate and re-process the different categories of reviewed items to generate the feedback metrics.

The validation method can be used to improve both precision and recall of searches. It can also be applied on existing collections of prior reviewed matters to improve overall ESI categorization criteria prior to the initial searches.

9.3 Effectiveness Benchmark for Searches

When comparing multiple search technologies, it is important to note that each search technology is likely to present different sets of results. Even within a single technology, multiple vendor offerings may produce different search results. Also, one has to evaluate the context of the search in the EDRM workflow to assess the effectiveness of searches.

a) In the case of searching meta-data using fielded search for Custodian Identification etc., there is a very high threshold of accuracy and that it is critical to understand variations in field properties, naming conventions and syntax.
b) In early case analysis, search needs to be fast, but results must be ordered by relevance so that search results can be evaluated quickly and iterated.
c) In the case of large-scale culling, searches must divide the population such that large populations fall into the culled/non-responsive bucket.
d) In the case of searching for potentially responsive documents, the iterative validation should eliminate false negatives and minimize false positives. .
e) In the case of potentially privileged documents, the searches should be targeted against the responsive population and have a low threshold for false negatives.

A significant consideration is whether a document is responsive and how that determination is made. It is quite possible that a document is considered responsive because of a subject-matter expert of an expert human reader familiar with the legal issues at hand has determined that the document is responsive. It is not necessary that a document that is responsive contain a specific set of search terms, so making the determination of responsiveness a subjective determination. Consequently, the notion of how effective a search has been in identifying responsive documents is itself subjective. Furthermore, it is often impossible to determine the human review based determination for the entire document collection, so a complete assessment of a search effort for every e-discovery undertaking is not feasible.

Similarly, effectiveness of a search methodology for privilege or confidentiality review is also difficult to measure. While these reviews typically involve fewer documents (i.e., only the responsive), the cost of a review escape is very high in that it may cause waiver of privilege. So, a search that has a few false negatives will likely result in inadvertent production.

In the absence of this, search effectiveness is often based on two methods.

a. Determine for a specific known collection, what the effectiveness of a particular search algorithm/methodology is.
b. Perform sampling on the collection and judge the effectiveness against the sample.

9.4 Search Accuracy: Precision and Recall

Search accuracy is often measured using information retrieval metrics Precision and Recall [3,4]. While these measures are good at recording effectiveness of a particular search, a complete e-discovery production may involve a combined set of searches, and a combined score is more relevant for discussion.

Precision measures the number of truly responsive documents in the retrieved set of responsive documents. Recall measures the number of responsive documents retrieved compared to the total number of responsive documents in the corpus. These two ratios have been used extensively to characterize effectiveness of information retrieval systems. [Ref. Blair-Maron ACM paper]

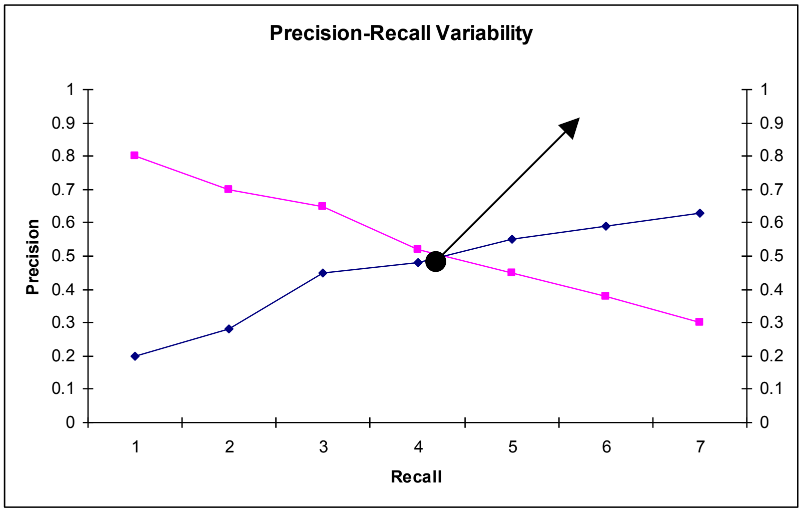
$$\text{Precision} = \frac{\text{Number of Responsive Documents Retrieved}}{\text{Total Number Retrieved}}$$

$$\text{Recall} = \frac{\text{Number of Responsive Documents Retrieved}}{\text{Total Number Responsive}}$$

A higher precision number implies that a large percentage of responsive documents are retrieved and only a small number of non-responsive documents were categorized as responsive. A higher precision number is helpful in establishing an efficient second-pass human review. Also, if precision numbers are high, one can even contemplate eliminating a second pass review.

A higher recall number suggests that the information retrieval system was effective in retrieving higher percentage of responsive documents, and fewer of the responsive documents are left in the unretrieved collection. A very small recall suggests improving the automatic retrieval methods.

Another factor that must be considered is the total number of documents. It is often possible to achieve high Precision and high Recall rates in a small corpus, but as the corpus size increases, these rates drop considerably. In many automatic information retrieval scenarios, the Recall rates can be increased easily, but the Precision rates drop. The intersection of the Precision and Recall point is a critical optimization data point, and information retrieval systems attempt to move this data point closer to the top right corner as shown below.



Another measure that is useful is a single measure known as the F-Measure, which combines both these measures into a single value.

$F = 1/(\alpha/P + (1-\alpha)/R)$

In the above formula, *P* is the precision ratio, and *R* is the Recall ratio, and α is a weight for giving different levels of importance to Precision vs. Recall. In most cases, equal weighting is chosen, so the α value can be set to 1. This single measure is representative of search effectiveness.

9.4.1 Determining Precision

For any search methodology, determining Precision is an easier task, since both the number of truly responsive documents and the number of retrieved documents can be determined easily. This is the case when the search selection criteria are specific enough to narrow the results to a small number. In our examples above, the selection criteria was able to narrow the total retrieved count to about 1% of the total size of the corpus. This set is then subjected to a human review to determine if the document is truly responsive.

In situations where the selection criteria are not sufficient to reduce the retrieved set, sampling technology is used to evaluate the precision. An example of a sampling based precision measurement is shown in section x.

9.4.2 Determining Recall

Since Recall measures the ratio of responsive documents against the full corpus, the number of responsive documents in the corpus is difficult to determine. This is because in general, effective automated culling methodologies leave a larger percentage of documents as not responsive. To perform a human review of the non-responsive collection is often cost prohibitive, and would defeat the initial purpose of the automation. To overcome this problem, we use sampling methodology to determine the number of responsive documents in the unretrieved set of documents, and estimate the responsive documents that were not selected.

An alternative to sampling-based evaluation of Precision and Recall is to rank the documents by a scoring order and then selecting only a certain number of top-ranked documents. This is often referred as relevance cutoff-based evaluation. Using a smaller set allows one to derive precision and recall, and then the results are extrapolated to the entire collection. We believe that this extrapolation is not as reliable as the sampling-based determination.

9.4.3 Iterative Searching

Quite often, retrieval effectiveness of a search methodology may be low, and it may not be apparent that significant number of expected documents are not actually located. As an example, a landmark study by Blair and Moran [8] determined that keyword-based searches had a recall of only 20%. Some of the ways to improve retrieval effectiveness are to iterate multiple times with newer search queries. These queries can be designed to improve both precision and recall.

Methods for improving recall are:

a) Supply additional search terms.
b) Introduce new search technologies, such as concept search.
c) Increase keyword coverage using wildcards, fuzzy specification.
d) Use the initial search as a "training step" and incorporate a learning mechanism to locate other like-documents.

Methods for improving precision are:

a) Supply more specific queries, involving more complex combinations of Boolean Logic to improve Precision.
b) Perform a pre-analysis of wildcards, misspellings etc., and eliminate unnecessary expansions of wildcards, thereby improving Precision.

9.5 Sampling and Quality Control Methodology for Searches

To evaluate effectiveness of a search strategy during early stages of case analysis is to perform sampling of the results and evaluate the sample. Proper selection of samples will likely yield quick, cost-effective evaluation which is critical when multiple evaluations are needed. Use of sampling is also helpful when performing final evaluation of searches as part of a quality control step. Once again, cost of evaluating large collection of ESI using a human review is an important consideration, and sampling reduces the number of documents that needs to be examined.

The basic application of sampling requires a random selection of items from a larger population and evaluation of the presence of an attribute (such as Responsiveness) in the sample, and then estimation of the characteristics of the population. In doing this, one accepts a certain *Error of Estimation*, and a *Confidence Interval* that the estimated measure is within that Error. As an example, sampling 1537 entries can provide an estimate of +/- 5% with a confidence interval of 95%. One of the results from sampling theory is that sampling a population for an attribute with a certain error and confidence interval does not depend on the size of the population. [Ref. SIX-Sigma]

An important aspect is exactly how one would select the samples. In general, a sampling effort takes into consideration broad knowledge of the population, and devices an unbiased selection. In most cases, the party performing the sampling has some knowledge of the population and there is one party with that knowledge. In contrast, most litigations where there is an adversarial relationship between a Requesting Party and a Producing Party, and since only one party has access to the underlying population of documents, agreeing on a sampling strategy is hard. An effective methodology is one that would require no knowledge of the data, but is still able to apply random selection process central to the effectiveness of sampling.

Additional details on various aspects of reliable sampling are described in Appendix-II.

Glossary

Boolean search	A search technique that utilizes Boolean Logic, using terms such as AND, OR, and NOT.
Concept search	A search technique that provides words which are similar in concept to a query word. A concept search will return documents that relate to the same concept as the query word, regardless of whether the query word exists in the search results documents. Concept searches can be implemented as a simple thesaurus match, or by using sophisticated statistical analysis methods
ESI	Electronically Stored Information.
Fuzzy search	A search technique that identifies ESI based on terms close to another term, with closeness defined as a typographical difference and/or change. For example, *snitch*, *switch*, and *swanky* can all match *swatch*, depending on how many incorrect letters are allowed within the search threshold
Inverted Index	An index that maps a keyword to the list of documents that contain the keyword.
Keyword Index	A technique that examines the ESI and builds a searchable electronic index. This index typically maps from a keyword to all the documents that contain the keyword.
Keyword Search	A very common search technique that uses query words ("keywords") and looks for them in ESI, using an index.
Privileged Documents	A set of documents that a Producing Party is not required to provide, since they fall into Privilege such as Attorney-Client Privilege. The existence of such documents should be recorded in the Privilege Log.
Privilege Log	A set of documents that a Producing Party did not produce on account of Privilege such as Attorney-Client Privilege.
Phrase Search	A search consisting of multiple keywords separated by spaces to form a single phrase. For a document to match this search, the entire phrase as entered must be contained within

the document.

Producing Party
A party that owns the complete collection of ESI, and is responsible for producing a portion of the ESI that is deemed to be relevant for a legal case or legal enquiry.

Proximity Search
A Proximity Search searches for multiple keywords. The matching documents must contain all the keywords, with the keywords occurring within a specified number of words from each other.

RDBMS
Relational Database Management System. This is a technical term for the class of software programs that manage data using a relational schema, such as Microsoft SQL Server or Oracle.

Regular Expressions
A pattern that describes what the search should return based on special characters added to the keyword. For example, car* uses the character * as a wildcard, and the resulting documents should contain words that begin with the characters “car”, such as *car*, *cartoon*, or *cartography*.

Relevancy Rank
A measurement of relevancy of a document, so that the Search Hits within a Search Results can be ordered. Relevancy measurements often involve counting the number of occurrences of a keyword within a document, as well as number of documents a keyword is found in.

Requesting Party
A party that does not own the ESI and is requesting that the Producing Party which owns the ESI to provide some subset of the ESI based on a Search Request.

Responsive Documents
A subset of ESI that matches the desired set of documents for the case.

Search Engine
A search component that implements the actual process of interpreting a search request and identifying subsets of documents. For example, a database management system such as Microsoft SQL Server contains a component that manages searches of the data stored in its databases.

Search Hit
A document in the ESI that is considered to match the requested Search Query.

Search Query
A well-formulated Search request that an automated search engine can interpret in order to produce matching results.

Search Results
A collection of Search Hits that match the intended documents of a Search Request.

Synonym Search
A synonym search returns documents that contain terms similar in meaning to the query words, usually using a thesaurus to determine which terms would match the query words.

Stemming
A search option that returns matches for all variations of the root word of the initial query word. For example, if the query word was *sing*, then if a search used stemming the search results would match *singing*, *sang*, *sung*, *song*, and *songs* as well as *sing*.

Tokenization
An operation that examines a document or block of text and breaks the text into words. Typically, a space is used to separate words, but special characters such as a hyphen, period, or quotation mark can also be used.

Truncation
A Search Specification that indicates that matching documents must contain words that begin with the letters entered, but that the matching words can end with any combination of letters.

Wildcards
Symbols such as * or ? included within a Keyword to indicate that the location where the symbols are used may match a single letter or multiple letters.

Appendix 1: Meta-Data Name Specifications

This section lists some metadata properties that one can use for specifying metadata search.

Metadata	Type	Description
AttachmentCount	Integer	Number of attachments
AttachmentName	String	Name of an attachment
Author	String	Author of a document
BCC	String	Blind-Carbon-Copy recipient
Category	String	Category of the ESI, such as EMAIL, DOCUMENT
Comments	String	Comments present in the Comment Field of the document
Custodian	String	Name of a custodian
DateLastAccess	Date	Date when Document was last accessed
DateCreated	Date	Date when Document was created
DateLastMod	Date	Date when Document was last modified
DateLastPrint	Date	Date when Document was last printed
DateReceived	Date	Date when document was last received
DateSent	Date	Date when document was last sent
DocumentID	String	Identifier of the document
DuplicateStatus	String	Whether the document has other duplicates.
Folder	String	Folder name where the document was found.
Source	String	Source of the document
EmailSubject	String	Subject Line of an email document
EntryID	String	For documents present in a Container, an ID within the Container that can be used to locate the document.
FileDescription	String	Description of the file
Filename	String	Name of the file
FileSize	Integer	Size of the file in bytes
From	String	For email documents, the sender of the document
MD5Hash	String	The MD5 Hash value of the document
ParentID	String	For documents present in a container, the EntryID of the parent.
Revision	String	Specific Revision String for the document
SHA1Hash	String	The SHA1 Hash value of the document
Title	String	The title of the document
TO	String	For Email Documents, the direct recipient of the email.

Appendix 2: Application of Sampling to E-Discovery Search result evaluation

As explained in Section 8, sampling is a useful technique for evaluating search strategy and the actual searches. There are several important aspects that the case team needs to consider when applying sampling for e-discovery purposes. Sampling, by its nature, produces results with certain margin of error as well as confidence measures. Therefore, the case team needs to use it carefully and consider its results in that context. Early case analysis for evaluating searches should use sample-based results as a direction to refine searches and is generally the responsibility of the case team. When sampling is used for final stages of documenting the production of ESI, the case team may need to communicate the results to the opposing team, with proper documentation of sampling parameters and sampling methodology.

This section explains various factors to consider during sampling.

A2.1 Sample Selection

A critical question is what exactly is the sample size to achieve a certain confidence level in your estimation. The theory behind sample size selection, error and confidence rates are discussed at length in statistics, a good starting reference point is [1, 2]. In some cases, the complete size of the ESI population is available to sample from, ahead of time, while in other cases the ESI population is a continuous stream of documents and we are not sure if we will have visibility into the entire population. In these cases, sampling requires selecting a document and randomly skipping a few subsequent documents in the stream. In other cases, the entire population is available, so sampling requires selecting/computing a random set of identifiable documents from the collection. Typically, one can use a pool of hash values of documents, bates number of documents and select some number of documents and only those documents are reviewed.

If your review is evaluating a single parameter for the entire collection (such as a document is Relevant vs. Not Relevant), sample size is governed by the mathematical formula:

$$SampleSize = \frac{1}{error^2}$$

If your desired error rate is 5% and your desired confidence level is 95%, SampleSize = 100/0.0025 = 400. Therefore, by examining one-in-four hundred documents, and determining the number of Responsive documents in that set, you can determine the actual number of relevant documents in the total population. The above formula assumes a normal distribution for the population, and the error rate measures values outside the two sigma.

Assume that the entire collection is 1,000,000 documents. Applying a requirement of 5% error rate, with 95% confidence level, you will be required to review 2,500 randomly chosen documents. Assume also that for the 2,500 documents, you determined 70 documents to be Responsive. If that were the case, you can make a statement that with 95% confidence, the total number of Responsive documents in the entire collection is in the range [26600, 29400].

Another example, this time, applying the sampling technology for measuring the Recall rate on search – let us assume that a search using Boolean logic over a set of search terms yields 5000 documents from a collection of 1,000,000 documents. One wishes to determine if the remaining 995,000 documents contain any Responsive documents. If you sample 2488 documents chosen randomly, and determine that there were 4 documents that were Responsive, you may determine that there are indeed potentially between 1520 and 1680 additional Responsive documents that could be present in the full collection. This statement again has a confidence level of 95%. As a quality improvement measure, you may examine the four documents and conduct a second search, with additional terms are specific to this query. Again, if an additional 3000 documents are selected using automated search, sampling may establish a smaller number of Responsive documents. In fact, you may increase your accuracy by selecting one in 300 documents (3,300 random documents) to get an error of 3% and confidence level of 95%. This iterative process produces a set of search terms with maximum Responsive documents.

Of course, the iterative expansion of search has the potential to retrieve larger number of documents than those that are truly responsive. In the above example, 8,000 documents were marked as Responsive by the two automated searches. If a manual review of this 8,000 documents establishes 7,500 to be truly Responsive, both your precision and recall rates are high.

In practice, the review may also be classifying documents as Privileged. For a second parameter, an independent set of sampling criteria can be established, to monitor its effectiveness.

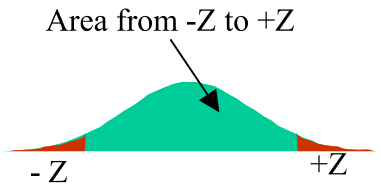
When the underlying phenomenon is not normal distribution, one has to utilize a different formula. As an example, studies have demonstrated that the inter-arrival time of events such as customers at a coffee shop or web search hits to a web site, or the amount of time it takes to render a page are Poisson-distributed. In the case of emails in ESI, the amount of time between the sender's submission of the message and the reply from a recipient, when viewed across all senders and recipients is also Poisson distributed [6,7]. A characteristic of Poisson distribution is the long-tail distribution. However, when the time of observation is very large relative to the inter-arrival time, Poisson distribution can be approximated to a normal distribution.

As an example, if you are estimating the number of responsive documents in a five year period by examining the responsive documents in a two-week period, the Poisson λ value is approximately 260/2 or 130. For this value, Poisson approximates normal distribution.

Sampling large populations involves a random selection of *n* items from a large population of items, and studying a characteristic on the sample and then using the results to estimate the prevalence of that characteristic in the population. In our application of sampling, we define a population as a collection of either the set of documents that are automatically selected as responsive documents (the retrieved set), or rejected as unresponsive (unretrieved set).

Once a random sample of *n* documents is selected, that sample is then analyzed using human review for responsiveness. If there are *r* documents in the sample that are truly responsive, the proportion of responsive documents in the sample is $\frac{r}{n}$ and is the sample proportion. Based on this, the proportion of the responsive documents in the population is estimated to be the same as the sample. The number *n* is the *sample size* and its selection has a bearing on the *error in the estimation* and the *confidence level* of the estimation.

To determine these measures, the number of different ways a random selection of *n* items is chosen is considered. Mathematical analysis estimates confidence interval and accuracy based on the assumption that if we were to plot the proportion of responsive documents of every possible sample, that proportion follows a normal distribution. Based on this, an error in estimation is selected to be within a certain number of standard deviations. The estimate of the proportion of responsive documents from a random sample can be stated to be within a specified number of standard deviations from the sample's proportion with a specific confidence level. The confidence level and error (based on number of standard deviations) measurements for a normal distribution are the *z-score*, which is a point on the normal distribution curve that encloses a certain area. For a confidence interval of 95%, the z-score is chosen so that the area under the normal distribution, between the –z and +z points total 0.95, and the area on the tails totals 0.05 of the graph.



The following table specifies various z-values for popular confidence levels.

Confidence Level	Area between 0 and z-score	Area in one tail (alpha/2)	z-score
50%	0.2500	0.2500	0.674
80%	0.4000	0.1000	1.282

90%	0.4500	0.0500	1.645
95%	0.4750	0.0250	1.960
98%	0.4900	0.0100	2.326
99%	0.4950	0.0050	2.576

According to this table, to achieve a confidence level of 95% and error of 5% indicates a z-score of 1.96. Given a certain z-score, we can then determine the sample size needed to achieve that error rate, using the following formulas.

Let *p* be the proportion of documents in the sample of sample size *n* that are determined to be responsive. This proportion and the z-value are related as follows.

$$e = z * \sigma$$

$$\sigma = {}^{1}\!/_{2\sqrt{n}} \quad \text{if } 0.2 < \text{p} < 0.8$$

$$\sigma = \sqrt{{}^{p(1-p)}\!/_{n}} \quad \text{if p} < 0.2 \text{ or p} > 0.8$$

For proportions that yield values in the range 0.2 to 0.8, the number of samples so that the error is *e* is given by the equation:

$$n = \left(\frac{z}{2e}\right)^{2} \text{..................(A)}$$

Similarly, for proportions that yield values in the range outside 0.2 to 0.8, the sample size is given by:

$$n = \left(\frac{z^{2}\, p(1-p)}{e^{2}}\right) \text{..........(B)}$$

One complication is that the sample size depends on the sample proportion, which has not yet been determined. To overcome this, we determine a running proportion *p* using sample size A. We then determine a new sample size using B using that proportion and we either increase or decrease the actual sample size. Decreasing a sample size would also mean that we stop review.

The following table specifies sample sizes for various error and confidence intervals.

Confidence Level	Error in proportion	Number of samples
50%	0.2500	2
80%	0.1000	41
90%	0.0500	271
95%	0.0250	1537
98%	0.0100	13526
99%	0.0050	66358

Two typical sampling scenarios are:

a) Confidence Level of 95%, with an error of 5% on the proportion (i.e. $p \pm 0.25$) requires 1537 samples to be examined.
b) Confidence Level of 99%, with an error of 1% on the proportion (i.e. $p \pm 0.005$) requires 66,358 samples to be examined.

An important observation is:

Sampling a population for a certain error and confidence interval does not depend on the size of the population.

The above result is extremely significant, since this implies that regardless of the size of the input data, we only need to examine the same number of documents to be sure that we have achieved a certain confidence level that our sample proportion is within a certain range of error.

As an example, if we have 100 million documents in the unretrieved set, we need to examine only 1537 documents to determine with 95% confidence that the number of responsive documents in the unretrieved set is within the margin of error. If we find that there are 30 documents that were responsive in the unretrieved set, we can state that we have 95% confidence that the number of responsive documents in the sampled set is between 28 and 32 (rounding up the document count on the high end, rounding down on the low end). Extending that to the 100 million population, we can determine that approximately 1,951,854 $\pm$ 97,593 are responsive in the unretrieved set.

In the case of a review where errors are expensive (such as review for privilege), 99% confidence with 1% error condition would require 66,358 samples. If we identify 200 privileged documents in such a sample, you will have 99% confidence that the number of privileged documents in the sample is between 198 and 202 privileged documents.

A2.2 Sampling Pitfalls

When utilizing sampling for either reducing the scope of collection and/or review, or for quality control, one has to be careful in the actual selection methodology used. This section illustrates some common pitfalls as a cautionary commentary.

Many of the instances of perceived failures of sampling are the result of incorrect application of sampling. As an example, the following are some failures.

a) The presidential election of 2000, where exit polling based on samples did not predict the outcome.
b) The election for California Governor, 1982, contested by Los Angeles Mayor, Tom Bradley
c) The 2004 general election in India, based on the India Shining campaign
d) The Democratic Primary Election of 2008, in New Hampshire

A2.2.1 Measurement Bias

Measurement Bias occurs when the act of sampling causes the measurement to be impacted. As an example, if police decide to sample a few drivers and estimate the average speed of drivers using the fast lane of the motorway by following the selected cars on a fast lane of a highway, the results would almost always be incorrect. This is due to the fact that most cars would slow down as soon as a police vehicle follows them.

In the realm of e-discovery, measurement bias could occur if the content of the sample is known before the sampling is done. As an example, if one were to sample for responsive documents and during the sampling stage, content is reviewed, there is potential for higher-level litigation strategy to impact the responsive documents. If a project manager has communicated the cost of reviewing responsive documents, and it is understood that responsive documents should somehow be as small as possible, that could impact your sample selection. To overcome this, the person implementing the sample selection should not be provided access to the content.

A2.2.2 Coverage bias

Coverage Bias can occur if the samples are not representative of the population due to the methodology used. As an example, if one were to device a telephone-based polling, the coverage is restricted to those reachable by telephone. The population that does not have telephones is not part of the sample, and if that population has a significantly different composition, that would not be captured in the result. This is true if there is a special correlation such as income or poverty levels and the presence of a telephone, and the sample-based polling is to estimate the level of poverty in a population.

In E-Discovery, such coverage bias occurs when large portions of ESI get excluded from based on meta-data or type of ESI. As an example, Patent Litigation may require sampling technical documents in their source form, and care should be taken to include these documents in the sample selection process.

A2.2.3 Non-Response Bias

Sampling errors may appear as a result of non-response Bias. The Indian Election of 2004 illustrates this phenomenon most dramatically. In this election, a large dejected population of voting public simply refused to answer exit polling requests, or were not literate enough to answer them. Also, this group of voters was not reachable by the mainstream media, and this group overwhelmingly voted in a way contrary to the projections from the literate voting population.

In e-discovery document review, non-response bias can occur if a large percentage of potential samples is off-limits for the sampling algorithm. As an example, if an e-discovery effort is identifying potential responsive engineering documents, and if the documents are in a document format and/or programming language that could not be sampled or understood, there could be a significant non-response Bias.

A2.2.4 Response Bias

The exit polls during election of Los Angeles Mayor Tom Bradley in 1982 indicated a response bias. These opinion polls were biased by the respondent's unwillingness to express their racist tendencies, so they chose to respond to the opinion pollsters in a way that did not evoke or expose their internal racist beliefs. In this case, the large voting block voted for the white candidate but stated to the exit pollsters that they voted for the black candidate.

Another form of response bias occurs when the participants in the survey are provided a set of questions worded in a certain way. If the same survey is worded differently, a different outcome is predicted. Also, certain wordings evoke an emotional response, tilting majority of the respondents to a Yes or No, and certain wordings overstate or understate an impact.

A response bias during sampling for e-discovery can manifest itself if sampling is used for large scale data culling. To guard against this, the sample selection process should avoid examining the contents of the documents. Any review of documents should be postponed to a post-sampling stage. A different type of response bias occurs during review, where the instructions and questions given to the reviewers and their wording can impact categorization of the reviewed documents.

A2.2.5 Sampling Time Relevance

In several cases, the selection of sample closest to the actual event is very significant. In the Democratic Primary Election of 2008, the sampling used for poll prediction was based on samples that were at least three days old, and did not include a significant number of late-deciders. Another exit polling sampling pitfall is using the earliest exit poll samples in order to be first to predict an outcome. This causes only the early voters to be counted, and the late voters are not counted to the same proportion. Also, using exit polling counts working public and absentee balloters disproportionately.

In the context of application of Sampling for e-discovery, there could be significant time periods when there was ESI and if these time periods are not properly selected, the final predictions may be completely inaccurate. As an example, if there was a significant corporate malfeasance during a certain date range, a stratified sample that treats that date range as more relevant should be considered.

A2.2.6 Sampling Continuous Stream

If the entire ESI collection is not available, one is forced to sample a stream of documents. A fundamental assumption is that the nature of the document collection does not change mid-stream. If this is not the case, samples taken at various points in time will reflect localized variations and will not reflect the true collection. Where possible, sampling should isolate collections into various strata and apply sampling within each strata independently.

10. References

[1] Statistical Sampling – Wikipedia, http://en.wikipedia.org/wiki/Sampling_(statistics)

[2] How to determine sample size: Six Sigma Initiative, http://www.isixsigma.com/library/content/c000709a.asp

[3] Amit Singhal, John Choi, Donald Hindle, David Lewis, and Fernando Pereira, AT&T at TREC-7, Proceedings of the Seventh Text REtrieval Conference (TREC-7), pages 239-252. NIST Special Publication 500-242, July 1999.

[4] Performance Measures in Information Retrieval, Wikipedia article, http://en.wikipedia.org/wiki/Information_retrieval

[5] Amit Singh, Google Inc., Modern Information Retrieval: A Brief Overview, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

[6] Anders Johansen, Probing Human Response Times: arXiv:cond-mat/0305079v2, February 2, 2008

[7] J.P. Eckmann, E. Moses and D. Sergi, Dialog in e-mail traffic, arXiv:cond-mat/0304433v1, February 2, 2008

[8] Blair, D. C. Maron, M.E. (1985). An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System, Communications of the ACM, 28, 289-299

[9] S. Deerwester, Susan Dumais, G.W. Furnas, T.K. Lansauer, R. Harshman (1990), "Indexing by Latent Semantic Analysis", *Journal of the American Society for Information Science* **41** (6): 391–407.

[10] Nicholas O. Andrews and Edward A. Fox, Recent Developments in Document Clustering, October 16, 2007, Deparment of Computer Science, Blacksburg, VA

[11] Naive Bayes Classifier and its use in Document Classification, http://en.wikipedia.org/wiki/Naive_Bayes_classifier