2013-12-09

# Enterprise Search Tactics for Expertise Location

Reinhart, Ian

UNIVERSITY OF CALGARY

# ENTERPRISE SEARCH TACTICS FOR EXPERTISE LOCATION

BY

IAN REINHART

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

DECEMBER 2013

**Abstract**

A discussion on the intricacies of enterprise search; contrasting web search, the systems distributed nature, various applications, and evaluation measures. Following this, a shift in focus to expertise search and the development of the Terrorist Expertise Locator (TEL), built using the latest open source enterprise search technology. Implementation follows an enterprise search methodology centered on tuning relevancy for the applicable dataset, in this case the Global Terrorism Database (GTD). Data analysis drives the application of information retrieval and information extraction techniques; stemming, query expansion, query operators, and document quality weighting. Evaluation of the enterprise search engine, in various configurations, provide insight into best practices used to improve relevancy. Three expertise calculation methods are compared; count-based, relevancy weighted, and expertise weighted. Validation of results are established by cross-referencing un-indexed popular media and government sources, including CBC news reports and FBI declassified files.

## Preface

I decided to undertake this topic for my master's thesis because of my domain expertise in enterprise search, I attained while working in the industry, in conjunction with my academic studies undertaken in the Database Lab research group, located at the University of Calgary, under the supervision of Dr. Reda Alhajj. I feel my unique combination of industry and academic experience, in a field that is not highly accessible to researchers because of its commercial nature, puts me in a unique position to present scholarly work in this area.

My enterprise search background in the industry started shortly after I completed my undergraduate degree in computer. I accepted an offer from an industry leader in enterprise search, Autonomy (now an HP Company). At Autonomy I operated in their engineering and professional services departments, where I implemented distributed enterprise search systems for several large corporations (Glaxo Smith & Kline, Deloitte & Touche, Bayer Pharmaceuticals, Sybase, IEEE Computer Organization, Synopsys). Now I operate an enterprise search consulting group, Logic Consulting Inc., and I continue to utilize the latest and greatest in enterprise search technology to satisfy my customers' needs.

Every search application I am involved in is unique in its inputs, users, and data, but many commonalities arise when evaluating a search application's quality. Perhaps the most important aspect in evaluation of search quality is the relevancy of search results. In this regard I will tend to evaluate my work.

**Acknowledgements**

I would like to thank my supervisor Dr. Reda Alhajj for his teaching and academic support. I would also like to thank my family and friends for their encouragement and help.

## Table of Contents

## List of Tables

## List of Figures and Illustrations

# Chapter One: **Introduction**

My thesis is a distinct contribution to the knowledge of enterprise search technology, and to the improvement of expertise search solutions. Originality of my work is achieved through the integration of enterprise search and expertise location. Incorporating both technologies provides a streamlined solution capable of determining expertise from authored work, there is no requirement for user profile data to generate expertise ranking using my design. Effectiveness depends on the correct application of search engine tuning, which needs to be driven by data analysis and iterative configuration testing, all of which will be detailed throughout this work.

## 1.1 Contributions

Major contributions of my work include,

1. Disseminating knowledge of enterprise search. Due to its commercial nature enterprise search research is not highly accessible.
2. A design and deployment process for a query-based expertise location system that can be integrated into existing enterprise search engines, making it consumedly attractive.
3. Providing novel best practices and performance measures to effectively tune enterprise search engines against unclassified datasets.
4. The first ever Terrorist Expertise Locator solution that can be used to search for expertise among terrorist groups.

5. A comparison of three expertise calculation methods; count based, relevancy weighted, and expertise weighted.

## 1.2 Problem Definition and Motivation

Enterprise search has immense practical importance in real organizations [7][8]. Companies that employ information workers and have a wealth of data commonly implement enterprise search systems to extract data-value and prevent employees from information overload. The Butler Group released a study in 2006 indicating that employees performing ineffective searches and wasting time looking for information cost companies 10% in salary expenses, as information workers spend on average 25% of their day searching for information to complete their tasks [13]. The lost productivity and costs associated to information retrieval tasks in the workforce warrants companies to implement search tools to exploit data-assets, identify opportunities, reduce risk, and gather insight [8]. Aside from cost cutting and productivity boosting search technology also enhances a company's ability to manage compliance and regulatory demands, enhance human-resource tasks, and upgrade the effectiveness of project management [8].

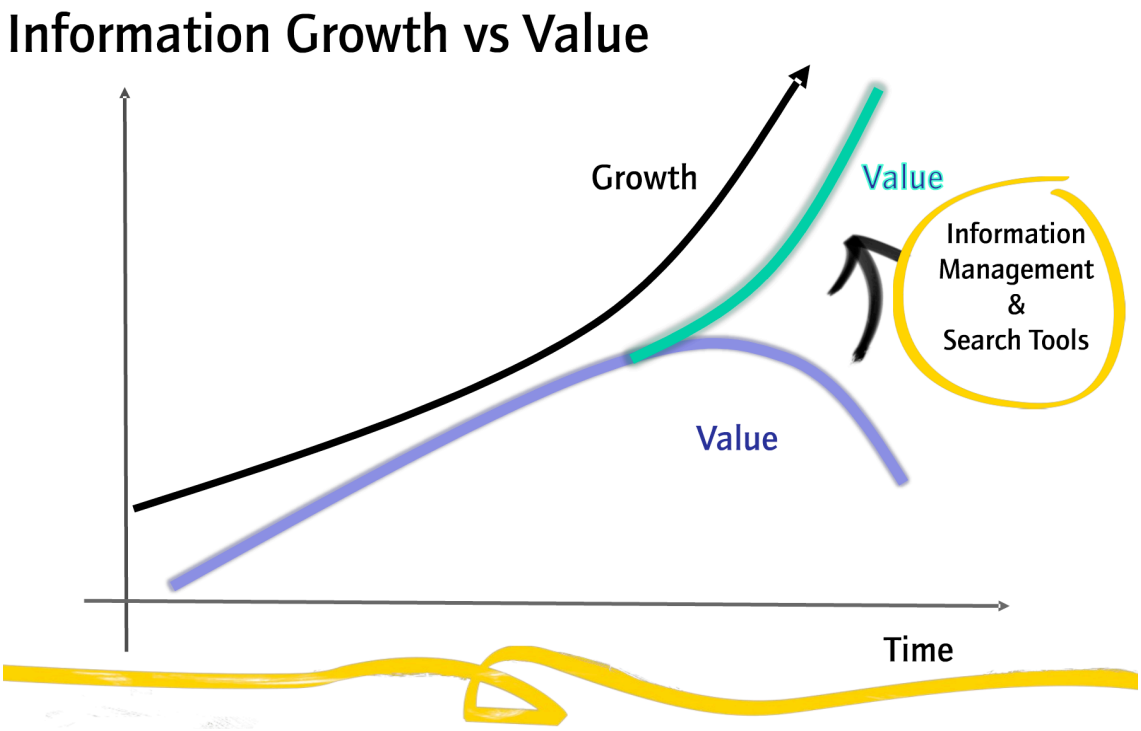



**Figure 1 - graph illustrating value gained from search solutions in the enterprise [80]**

Expanding on the idea of information growth vs. value and where the need for search tools fit, picture your email inbox. If you never delete emails (like myself) you could spend days looking for an important message, but with a search toolbar like the one built into Gmail or the iPhone you can quickly retrieve relevant messages in seconds. Likewise, in the enterprise when an employee quits, is fired, or retires their workings and knowledge can be lost unless it is indexed and made searchable.

Enterprise search technology aims to bring the best information sources to a user. Take for instance the devaluation trend where people go to Wikipedia for information instead of the best available sources [10]. Enterprise search technology can

be used to reverse this trend of devaluation by creating search applications that integrate the best sources for search.

One application for enterprise search that is gaining popularity in the industry is expertise search or people search [1]. Working in the industry I have seen many interested clients requiring such technology to help manage their professional service resources. In this section I will try and answer what is expertise search, why it is important, and how it can be best implemented using today's enterprise search technology.

Expertise is a matter-based opinion founded on the consideration of known experts. One could further define expertise as a human characteristic, claiming an expert as one who performs a task better than his peers. In most industries today expertise is established by what others say. This granting of expertise by others often leads to the promotion or demotion of human resources, and the roll-on effect of expertise decision making can be detrimental in its impact to business performance.

So because human expertise is valuable, perhaps the most valuable resource in the world, it is important to be able to assess it and locate it. In today’s corporate environment finding expertise has relied mostly on human resource personnel. These individuals use social and collaborative processes, such as making phone calls to others who may have insight, following up on references from a prospect, or combing work-related social websites like LinkedIn [50]. Quite often these processes do not take into account all available information, and the result can lead to poor decision-making.

Over the last twenty years corporations have ramped up on technology to help drive their business [7]. System implementations may vary, however most large

corporations have email services, fileshares, websites, relational databases, and content management systems. The information contained within these systems is vast, and analysts suggest its content will grow over 800% in the next 5 years [5]. The content contained across these data-sources can be used by expertise search engines to locate personnel for a particular task. Consider the need of a large consulting firm who requires experts in a particular field for an upcoming project, or a doctor looking for a specialist to treat a patient requiring a skill beyond his own. An expertise search engine with an index of all personnel and their textual contributions, which may be in the form of documents, emails, and website postings, may be an invaluable tool to help locate the best individual for a specific task.

### 1.3 Proposed Solution

Expertise search solutions tend to focus on expert finding in specific domains, extracting representations from known document types [1]. Initial designs employed a database containing skills and knowledge of each individual in the organization [32]. The skills database would be manually constructed and queried, requiring considerable effort to setup and maintain. Automated approaches have been devised to mine data repositories of organizations and implicitly build profiles for each individual [26][29][30]. However, this approach is resource intensive, time consuming, and not capable of handling real-time queries.

As enterprise solutions are developed in industry, details are hard to come by, and it is my desire to detail the construction of a query-based expertise locator that

overcomes the fallbacks of previous designs. Applying my knowledge of enterprise search I will drive the solution using the latest open-source enterprise search engine SOLR [11], infusing it with custom configurations in order to tune the relevancy to the applicable dataset; containing a mix of structured and unstructured content.

### *1.3.1 Expertise Locator*

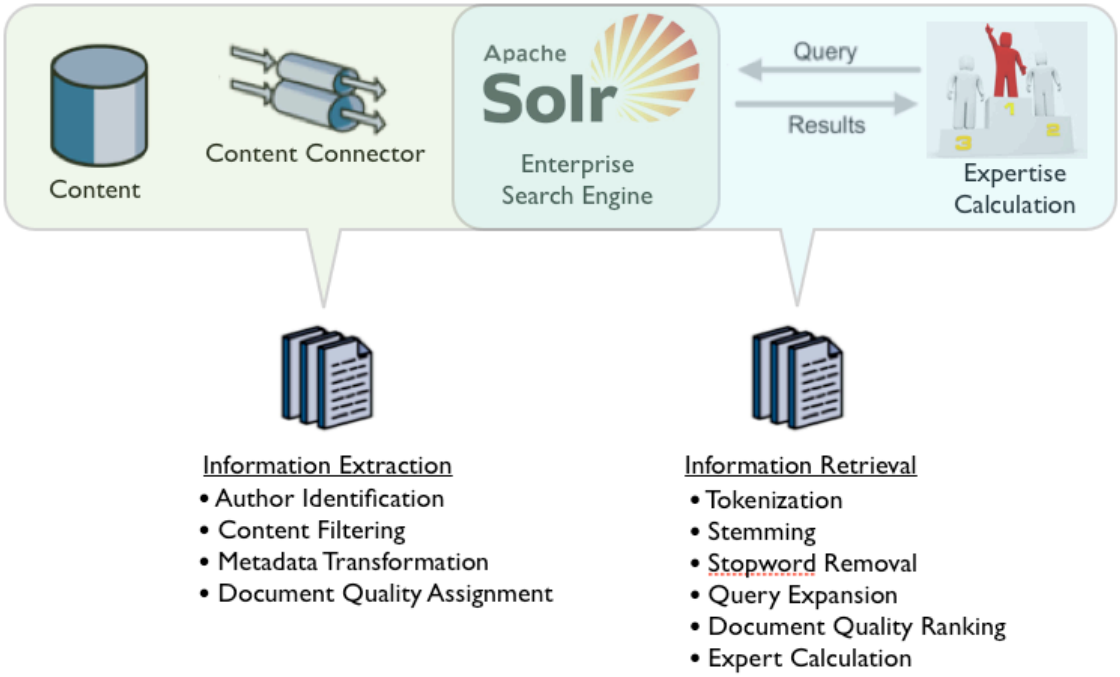



**Figure 2 - high level overview of proposed expertise locator**

The solution will be designed in a service-oriented architecture, with each component providing a discrete piece of functionality. Content will be fetched from a connector, which will perform a range of information extraction techniques to enrich the data with information to assist with expertise determination, as well as transform and filter content to improve search ability. SOLR will provide enterprise search capabilities, and will be configured to utilize a range of information retrieval techniques

to achieve the highest possible relevancy for the content that is indexed. An expertise calculation will be applied to search results, and an expertise ranking will be returned for a query.

# Chapter Two: **Background and Related Work**

The following chapter will cover knowledge related to enterprise search, information extraction and retrieval technology, performance measures, and expertise location.

## 2.1 What is enterprise search?

Enterprise search considers information workers whose task in an organization is aided by searching data to complete a specific task [1][2][3][7]. Search technology provides information extraction (IE) and information retrieval (IR) services across a variety of data repositories present in the enterprise [1][2][3][15]. Typical sources of content include relational databases, intranets, email servers, file-shares, and content management systems [1][7]. Enterprise search systems also typically utilize secure search technology to control access to information and the enterprise users [9].

## 2.2 Enterprise Search Vendors

Working in the industry I found the most common enterprise search vendors considered for use in enterprise search solutions to be Autonomy [79], Google [19], and Apache's open-source alternative SOLR [11].

Autonomy was founded in the UK in the mid-nineties, and was the first search product to become available of the three. Over time Autonomy expanded its products and services acquiring main competitor, Verity, and other specialized enterprise software companies like Zantaz and Interwoven. In 2011 Autonomy was purchased by

HP for $10.2 billion [45]. Autonomy's flagship product IDOL (Intelligent Data Operating Layer) is the search engine component. Its processing of information is referred to by Autonomy as Meaning-Based Computing [44]. Autonomy is seen as the Cadillac of search technologies, providing the highest level of search customization through hundreds of configuration settings.

Google released its first enterprise search product, the Google Search Appliance in 2002 [46]. One of the differentiating factors of Google's product was that it bundled their enterprise search software with Dell manufactured hardware [47]. Their appliances are marketed to be capable of searching 10-30 million documents in a single appliance, depending on the model purchased [48]. Appliances can also be connected to expand indexing capability beyond 30 million documents. Google's GSA offers the enterprise a powerful and simple plug and play search device.

SOLR the open-source alternative to enterprise search, includes many popular features of its commercial counterparts. SOLR uses the Lucene search library, another Apache Foundation project, to provide search and indexing services. SOLR originated from a CNET search project that took place in 2004, and in 2006 its source code was openly published and donated to the Lucene project [49]. SOLR's flexible external configuration and plugin architecture make it an ideal choice for search projects requiring a high level of customization.

## 2.3 Enterprise Search vs. Web Search

The key difference between enterprise search and web search is the material being indexed and how relevancy is calculated [7]. It is common for websites to be indexed in both cases, but enterprise content encompasses many other types of data such as office documents, spreadsheets, slide presentations, email, database tables, and content management systems like SharePoint or Documentum.

The algorithms for determining relevance of search results are also very different between enterprise search and web search [7]. Web search uses link structure to assign authoritative ranking to web-pages, see PageRank algorithm [16]. Enterprise search does not have the luxury to use link structure when calculating relevancy because it deals with content that does not include this information, or because there is a need to aggregate content without any link structure. The majority of enterprise content is usually in an unstructured form, for example word documents, pdfs, and presentations [7].

Instead of using link structure to determine relevancy enterprise search technology tries to use other tactics such as field weighting, term occurrences, term distances, document freshness, and advanced query operators to filter out irrelevant documents [3].

## 2.4 Data Integration

In enterprises data resides in multiple repositories and there is a need for enterprise search applications to search across content in all locations [7][12] [14]. For

example *Company A* desires a search tool for litigation purposes and requires the ability to search employee email and office documents. The email and office documents reside in different locations and the content is stored in different formats. Enterprise search technology overcomes this problem through data integration into a central search appliance. Search providers offer "Connectors" for popular data repositories to feed the search appliance with content. The connectors are responsible for continually updating the search appliance and transforming the content in order to provide a unified search experience.

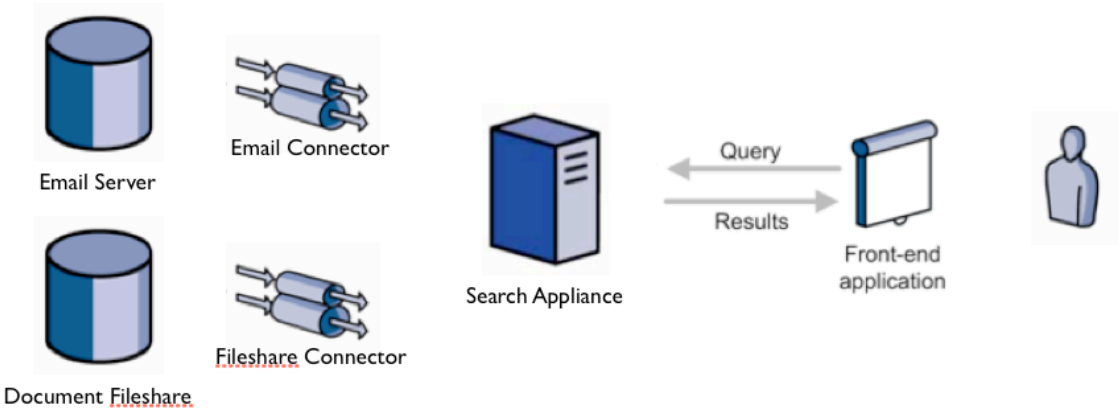



**Figure 3 - tightly-coupled search architecture**

This tightly-coupled architecture is great for enterprise search applications because it allows data to be combined under a single searchable schema. This greatly increases performance because a single query language can be used to search all types of data (email, documents, etc.). However some issues arise around the quality of the data being searched because it is difficult to keep the search appliance in sync with all the data repositories. Secure data repositories pose another concern with the tight-coupled search architecture because it is possible a user could access secure content on

the search appliance before an access control change to prevent the user from seeing it is mitigated to the search appliance (in other words the security polices between search appliance and data repository go out of synch for a period of time). This concern can be addressed, at the cost of search response time, in an early-binding security model, which will be discussed shortly.

Data integration linkages are key to enterprise search applications in order to allow aggregation of content residing in multiple sources into a single document [3]. This merging of information from multiple sources is often useful to enable faceted search functionality and provide or update metrics used for sorting and filtering search results [3]. For instance many companies keep structured information on a database engine and unstructured information on fileshares. Typically the database will contain a field that links the structured information to the unstructured, and through this link search can use the structured information for filtering/browsing/sorting and the unstructured for phrase matching.

### 2.5 The Process Flow of Enterprise Search

Enterprise search systems of late are designed in a service oriented architecture (SOA) [1]. Each interoperable service is responsible for fulfilling a phase in the general process, from ingestion to querying.

Usually enterprise search systems are standalone systems, in the sense they encompass a copy of all the data to be queried. This allows queries to execute timely

across a variety of data repositories. For secure search, user/group information may be contained in the enterprise search system to improve secure search retrieval speeds.

The process flow of enterprise search systems can be broken up into 4 broad phases. In chronological order these phases are ingestion, transformation, indexing, and querying. Ingestion involves identifying the data repositories that are required for search, connecting to them, and extracting the content and metadata contained in the repository.

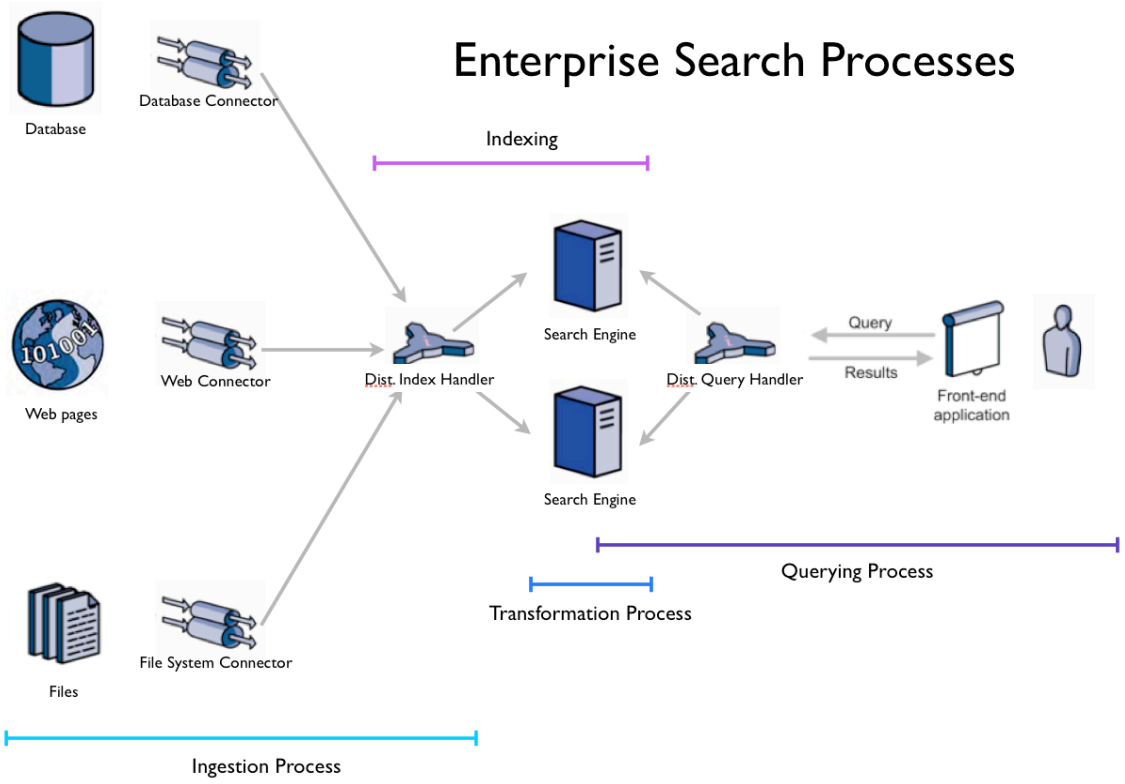



**Figure 4 – enterprise search processes**

### *2.5.1 Ingestion*

The components required to perform ingestion are commonly known as connectors. Key enterprise search systems provide connectors to retrieve content from

websites, fileshares, and relational databases [11][79][19]. Connector frameworks are also available to facilitate developers to write their own connectors for accessing proprietary or customized data repositories [81]. Connector frameworks provide common connector functionality like scheduling, batch processing, authentication, and content filtering.

The ingestion phase starts off with iterative runs of connector services in order to get the output desired. Connectors typically create output in a structured xml form, however some commercial enterprise search systems use proprietary output formats. Connector output is delimited at a document or record level. For example, the output produced from a fileshare connector would be delimited per file and each file would contain content and metadata pertinent to the file. When connectors have been successfully configured a full crawl of the data source is kicked off to collect all content. After the initial full crawl of content collection completes future content deltas are picked up by the connectors' ability to either pull or push changes depending on the connector's model.

In the case of large enterprise search systems content must be distributed across multiple search engines. In such cases a distributed index handler is deployed between the connector and search engines (see figure 4 above). The purpose of the distributed index handler is to segregate content evenly across the search engines. The use of the distributed index handlers can increase indexing speed and query speed when the size of content to be indexed is large enough to warrant distribution. In the industry I have noticed distributed index handlers are commonly used when the number of documents

to be indexed approaches a number greater than 2 million; however every search technology has a best practice that may suggest other document counts.

### *2.5.2 Transformation*

The transformation phase encompasses all work performed on the connector's output prior to indexing. This includes manipulation of content and metadata, information extraction, and content aggregation. Manipulation of metadata is an important task to consider if search is performed across a variety of data sources. Advanced search features, such as faceted search, will require field mapping effort to ensure filtering can be applied across all content types [3]. For instance, if a company wanted to discover email and document content from a particular author, it would be necessary to map the author fields to a common fieldname.

Information extraction (IE), another sub-process of transformation, is the extraction of knowledge mainly from unstructured sources [2][15]. Essentially entities like people, places, and things are deduced from unstructured text [15] and placed into new metadata fields in the primary document. IE can be performed on the unstructured text of enterprise content through the use of entity extractors [2][15]. Entity extractors can be as simple as a lookup list for identifying State/Province abbreviations, or as complex as a social network analyzer used to identify relationships [2]. The entities extracted from IE processes can be used to facilitate document clustering, document classification, and improving faceted search [2].

Content aggregation is the process of combining related content from more than one source. As an example consider email search where content becomes an aggregation of email and attachment(s). It is also possible that documents may have external metadata living on a different data source. This is quite common nowadays with the advent of content management systems in the enterprise. In such cases the external metadata is fed to the search appliance using a special type of connector. The process requires some link between the external metadata and the primary document. To give you an example picture an ecommerce site that allows you to sort products based on a "most popular" metric. The "most popular" metric would be the external metadata and the primary document would be the product description. The external metadata requires a connector to extract and merge the information into the primary document to support this type of functionality.

### *2.5.3 Indexing*

Resulting output from the transformation phase is parsed and stored in an index optimized for fast lookup and accurate information retrieval, this is considered the search engine component. Optimization is achieved partly by storing only important terms and characters. Unimportant terms like "the", "and", "to" are omitted thru the use of a stopword list, and help reduce the size of the search index and increase retrieval performance. Punctuation characters are also typically ignored during indexing to help further reduce the number of unique terms to be stored.

Some principal considerations in designing a search index include decisions on real-time or scheduled indexing of datasources, sectioning of documents, tokenization, stemming of terms, and metadata inclusion. Real-time indexing requires significant processing power when data is abundant and rapidly changing. A more economical approach is to allow for some delay in synchronizing datasource content to the search index via scheduling indexing tasks. Document sectioning can improve information retrieval in large documents by returning the most relevant section to a user. Tokenization splits text into tokens, treating whitespace and punctuation as delimiters (delimiter characters are typically discarded) [82]. Stemming is the process of reducing words to their stem or base form and treating the set of terms as synonyms [82]. For example the terms “searching” and “searched” stem to their base form “search”. Stemming algorithms are applied during indexing to enhance discovery of relevant texts. This way if a user searches for the phrase “spider-man movies” they will get results that also include the terms “spider man”, “spiderman”, and “movie”. Selective inclusion of metadata to a search index also improves information retrieval and discovery of relevant text. Exposing metadata is also a pertinent requirement for implementing faceted search [3].

### *2.5.4 Querying*

The final stage, querying allows users to retrieve relevant indexed content. Often users access search through a portal; a webpage where they can issue their queries. Common query options include matching terms and phrases, Boolean operators,

filtering options (i.e. date filtering), and sorting options. More advanced query options that may be available to users include spellchecking, highlighting contextual snippets, fuzzy matching, wildcard matching, faceted search, proximity search, and predictive search completion.

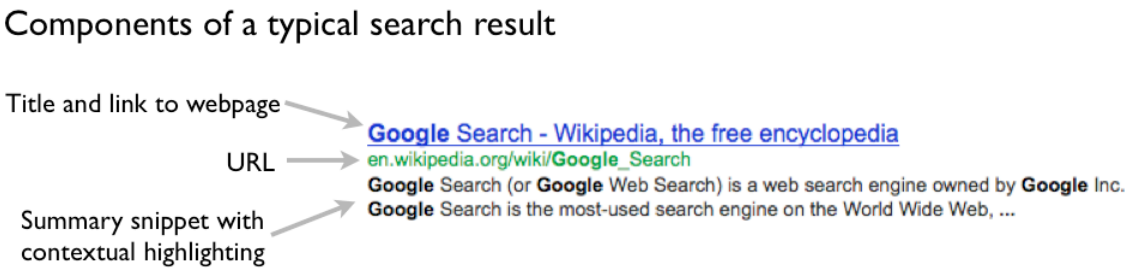



**Figure 5 - components of a typical search result taken from Google.com**

After the query is issued from the webpage it is sent to the search engine for processing. The search engine will execute the query against its stored index and return the results to the web application for display. Often results display a document's title, filetype, location or link back to native document, important metadata (i.e. price, color, weight), and a contextual summary snippet highlighting the queried terms as well as the query's synonymous terms.

By default results are sorted by their relevancy score. Relevancy scores are calculated using query term occurrences found in indexed content [83]. Enterprise search systems also allow for relevancy tuning, such as document freshness (newer documents score higher), and field weighting (term hits in certain metadata causes a higher or lower relevancy score). For example, most enterprise search solutions consider a search term occurring in the title of a document to be more significant than if the term occurred in the body of text.

## 2.6 Information Security

Information security protects sensitive content from unauthorized access. The common goals of information security are protecting confidentiality, integrity and availability of information [20]. The majority of enterprise search applications are heterogeneous [21], meaning they can contain several different security models that have to be integrated within the search application. Because of this complexity it's important to understand the options available.

Security modules for enterprise search applications try to provide integrated security solution's that match your data repositories native security type. Securing the front-end of an enterprise search application is accomplished by authenticating users on login. Single Sign-on protocols, like Kerberos, are popular methods of authenticating into an enterprise search application because it allows a user access to several secured areas with a single sign on. It would be very painful to have a user enter credentials for each repository before retrieving search results. After a user is authenticated the search application holds onto the user's role and group information to security trim results.

Enterprise search security methods typically provide early and late binding models. Early binding works as follows. A user queries the search application. The application then checks the security entitlement of a user in real time at the original data repositories of documents that are returned.

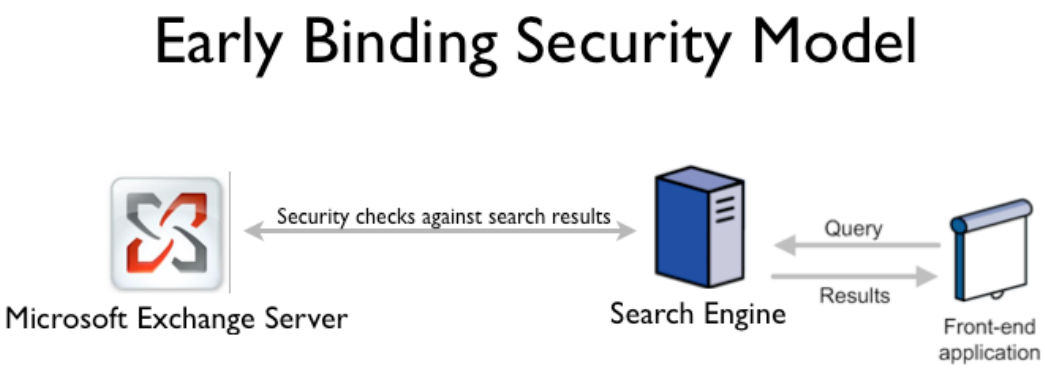



**Figure 6 - early binding security**

The advantage of this early binding approach is in the freshness of the security information it provides. This method ensures the security information used to trim results is up-to-date. The disadvantage of this approach is speed. The search application has to go out and authenticate the user against each native repository's security system. Due to this requirement the response of the search application will be negatively affected. This security model is suitable for environments where security entitlement for documents changes frequently, or freshness of security is more important than search response time.

The streamlined option for securing search content is the late binding model. The late binding model requires that documents' security information is indexed during the ingestion of the data. Usually this is accomplished via the connector service that is used to extract the content to be searched. The connector will store the security information along with the document's data. The security metadata for the document is commonly stored in an access control list (ACL) format. A document ACL should specify the users and groups allowed to access it, along with the users and groups that

are denied access to the document. It is also common for the document ACL to be encrypted to protect the access control information from prying eyes.

When a user logs into the search application the user's roles and groups are retrieved from their native sources, like LDAP,  or from a user/group server that operates to keep security information up to date and synced across all security types.. When a user queries the search application, it compares the user's roles or groups against the ACLs in the result set to determine whether the user is permitted to view the documents that match the query.

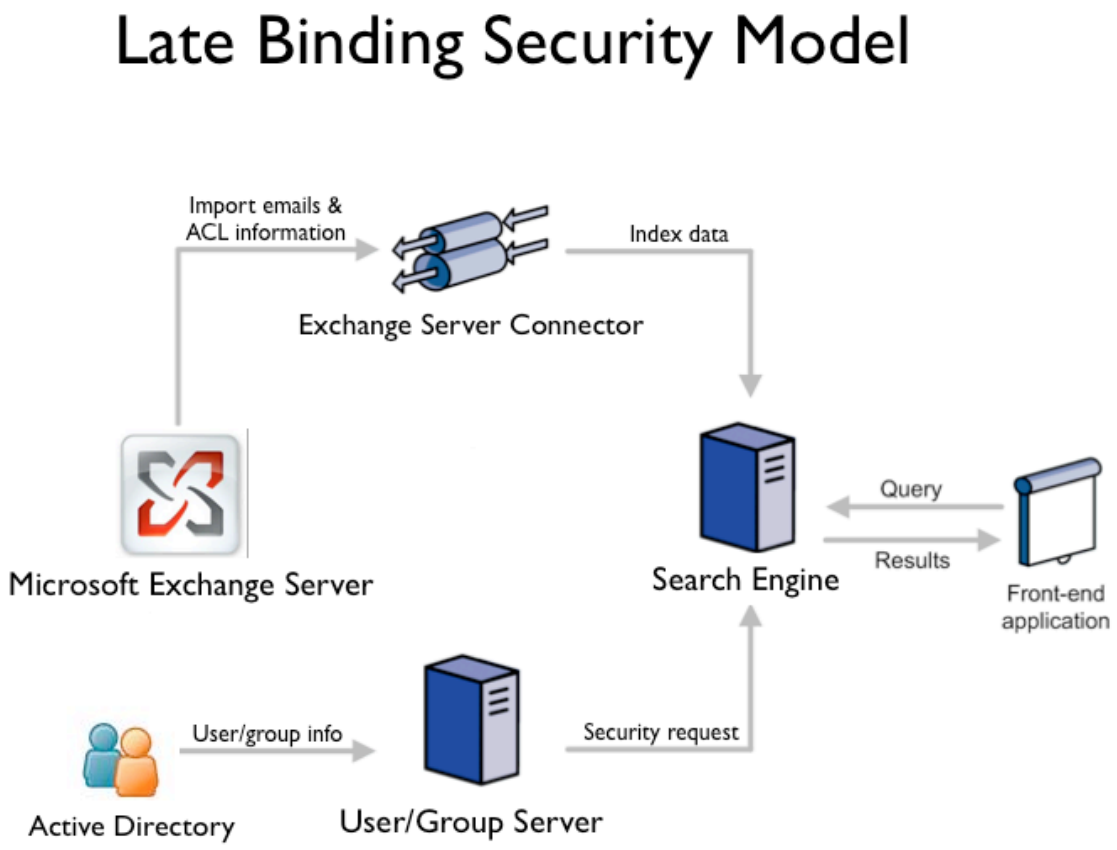



**Figure 7 - late binding security**

This model has a significant speed advantage over the early binding model. The speed is achieved by internally storing all user/group information and document ACL. Allowing the application to search content without the need to connect to the original data repository to check the security entitlement for each document. The disadvantage is that there is the potential for delay between the security settings changing in the original data repository and the information updating in the search application. Due to this the late binding security model is best suited for environments where security entitlement for documents does not change often, or search performance is of greater importance than security freshness.

Enterprise search applications can inject their own security settings at a level higher than the document. This can be referred to as role based security, where access can be granted to certain sections of content that span across multiple repositories and security types. Security information can also be generated for content that does not have a native security module in place. For example, patient health information in an Electronic Medical Record system may not have a native security module, but the system still requires secure search, and so a custom security module is developed to secure the content.

Securing communications between the distributed components of an enterprise search application is also possible and achieved through secure socket layer (SSL) communication [84].

## 2.7 Major Types of Search

In order to create a successful enterprise search solution the user's intent and the contextual meaning of the enterprise content must be understood. This knowledge will help determine the type of search functionality that will be required to generate relevant results.

### *2.7.1 Navigational Search*

When an information worker issues a query with the intention to bring back a specific document, that search can be classified as navigational [23]. There are many scenarios that require navigational search. Take for example lawyers searching through a client's email server for specific terms of importance in a legal matter. Or more commonly, when someone issues a query on Google to return a specific webpage.

This type of search relies heavily on exact word and phrase matching. Other more advanced functionalities include faceted search capability, date ranking, query term spell-checking, highlighting, intelligent summaries, and type ahead suggestions.

### *2.7.2 Faceted Search*

From the user perspective, faceted search (also called faceted navigation, guided navigation, or parametric search) breaks up search results into multiple categories, typically showing counts for each, and allows the user to "drill down" or further restrict their search results based on those facets [2][3]. A good example of faceted search can be seen on popular online retailers, like shoe department store Zappos [85] or Amazon

[86], where users can navigate their inventory via numerous filters like brand, style, colour, size, and price.

### *2.7.3 Conceptual Search*

Enterprise search technology can answer questions and find information conceptually relevant to a user's query terms. In this regard users do not necessarily need to know what they are looking for to find the information they seek. The processes enabling conceptual search include tokenization, stemming, clustering, classification, and synonym expansion.

#### 2.7.3.1 Tokenization

Tokenization is the process of breaking textual content into words or other meaningful elements called tokens [82]. The list of tokens becomes input for further processing such as stemming [82]. Typically, tokenization occurs at the word level, often relying on simple rules,

- All contiguous strings of alphabetic characters are part of one token; likewise with numbers.
- Tokens are separated by whitespace characters, such as a space or line break, or by punctuation characters.
- Punctuation and whitespace may or may not be included in the resulting list of tokens.

For example consider the following stream of text,

"Please, email john.doe@foo.com by 03-09, re: m37-xq."

A Standard Tokenizer, found in open-source search application SOLR [11], would use the following rules,

- Periods (dots) that are not followed by whitespace are kept as part of the token.
- Words are split at hyphens, unless there is a number in the word, in which case the token is not split and the numbers and hyphen(s) are preserved.
- Recognizes Internet domain names and email addresses and preserves them as a single token.

And the resulting tokens would come out as,

"Please", "email", "john.doe@foo.com", "by", "03-09", "re", "m37-xq"

#### 2.7.3.2 Stemming

Stemming is the process of reducing terms to their stem or base form [82]. It is useful in enterprise search because it increases recall across the searchable content by allowing search hits to come back for any term's stem. For example, consider the words *loved, lover, loving, and lovingly* can all be reduced to their base stem *love*.

There are several stemming strategies, but the most commonly used is Porter's stemmer [22]. Porter's stemming algorithm uses reduction to stem a word to its root. It is a rule based algorithm that does not require a dictionary [22]. The algorithm is too intricate to present here, but I will go through it at a high level. If you are interested in the full algorithm you can find it at the official Porter Stemming site [42].

Porter's algorithm consists of 5 phases of reduction. Each phase selects a rule that applies to the longest suffix of a term. Here are some examples,

| Rule | Example |
|---|---|
| SSES → SS | caresses → caress |
| IES → I | ponies → poni |
| SS →SS | caress → caress |
| S→ | cats → cat |

[22]

To prevent terms from getting cut too short there are checks on length. For example, (*m* > 1) EMENT →, would map *replacement* to *replac*, but not to *cement* [22].

2.7.3.3 Clustering

Clustering in enterprise search is the act of grouping similar documents together [3]. The end goal is to have documents as similar as possible within a single cluster, and documents as dissimilar as possible between clusters [3]. Clustering is the most common form of unsupervised learning [22], meaning no expert decided on the classes of each cluster. Rather the makeup of each cluster is determined by the content of the document set because clustering puts documents together that share many of the same terms.

Clustering can be useful in enterprise search applications to suggest or recommend documents that a user is interested in by simply pointing the user at other

documents in the same cluster. Clusters can also be used to group search results and display the top hits from each cluster, giving the user a broader spectrum of results by preventing the case where all hits come from a single cluster.

Clusters can also be used to guide or expand a user's query to include commonly occurring terms in the cluster that were not issued in the original query. Often this is implemented on user interface as a cloud view.

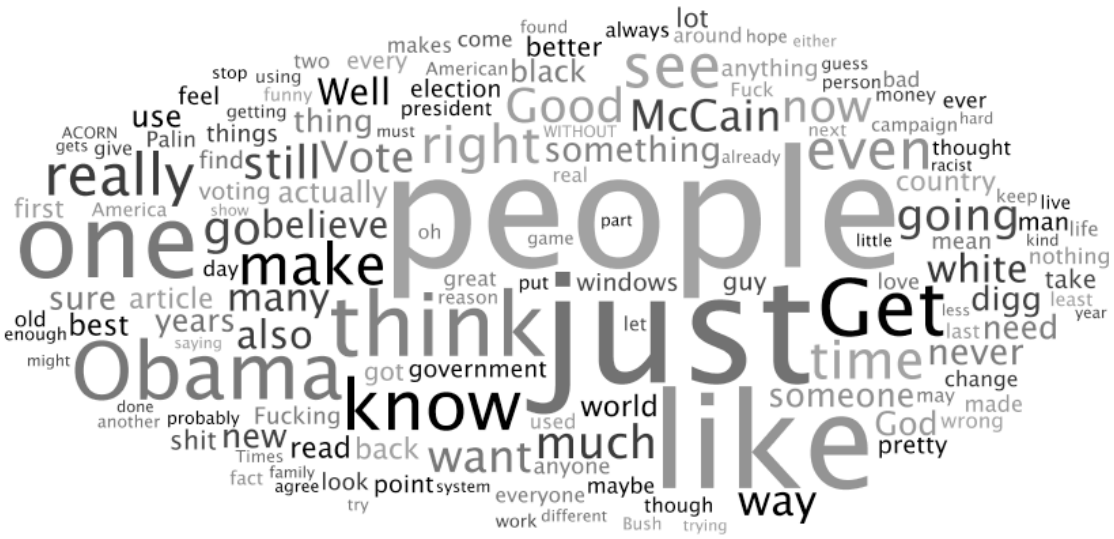



**Figure 8 - textual cluster visualization [87]**

2.7.3.4 Classification

In enterprise search classification refers to categorizing and labelling a document into a specific set of pre-defined categories based upon a document's textual content [22]. For example, a medical classifier may take the textual input of a clinical record and try to classify it into a medical sub-discipline such as radiology or

cardiology. Manually classifying documents is too expensive in enterprise applications, so automatic classifiers are required.

Automatic classifiers require rules to classify a document [22]. The rules can be written as Boolean expressions that capture a certain combination of free-text terms and or metadata values to determine a specific category. For example the following student metadata could be used to construct a Boolean expression to classify exceptional students, Exceptional = (Attendance > 0.9 AND GPA > 3.7 AND Publications > 5)

Manually created rule sets are effective, but also labour intensive. Machine-learning based classification allows the rule set to be determined automatically by using training data. This method requires a set of pre-labelled training documents for each category. This style of learning is denoted as supervised because a human is required to aid the learning process, by providing the training set and categories ahead of the classification process.

Text classification using a supervised learning approach can be formally described as $\gamma : X \rightarrow C$ , where X is the document space, C is a fixed set of categories, and the supervised learning method is $\gamma$ [22]. The learning method can be broken down to $\Gamma(D) = \gamma$, where the training set D is used as input and returns the learned classification function $\gamma$ [22].

Take the following example below where we have hierarchical categories regions (UK and China), industries (poultry and coffee), subject areas (elections and sports), training sets for each category, and uncategorized document *d.* The classification function assigns the new document "first private Chinese airline" to class $\gamma(d) = China$, the correct assignment.

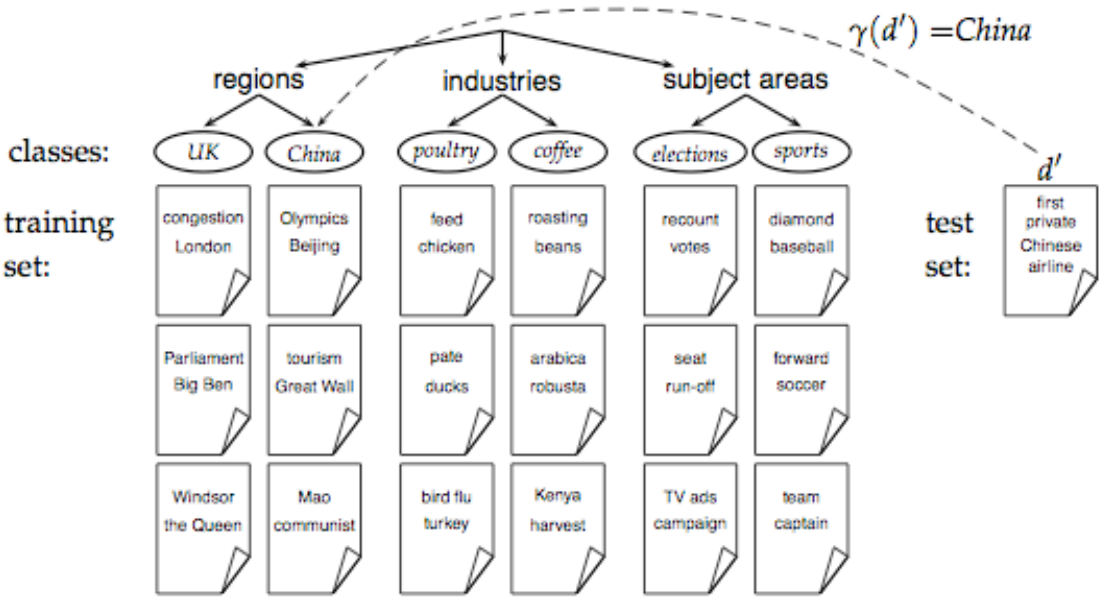



**Figure 9 - example text classification [22]**

2.7.3.5 Synonym Expansion

In many domains a concept can be referred to by several different terms. A grouping of terms that imply a similar meaning are synonyms. Synonymy has a great impact on the recall of enterprise search applications. For example, if you searched for *car* you would want it to bring back documents that contained *automobile*, *auto*, *sedan*, and possibly *van* and *truck* too. Users of a search system can try to overcome synonymy themselves by entering all the possible synonyms in their query, however there are methods to perform this type of query expansion automatically.

Query expansion can be performed automatically with the use of a thesaurus. Generally the thesaurus is presented in a flat file with each line representing a specific

concept. For example the synonym for cat may look like cat, feline, mouser, puss, pussy, tabby, etc.

### *2.7.4 Collaborative Search*

Enterprise search applications that utilize collaborative search allow users to work together in their information retrieval activities. Users have the ability to tag documents, essentially enriching the document's metadata with user generated comments. Collaborative search has many benefits enabling greater search recall through synergistic efforts, improving users search skills through exposure to others search behaviour, and an opportunity to strengthen social connections [24]. Collaborative search can also be used to influence relevancy by allowing users to vote on a document's significance, similar to how the popular website Reddit propagates hot articles to the top of its front page [43].

### *2.7.5 Semantic Search*

Semantic search aims at improving relevancy of search by trying to understand the searcher's intentions, and the context behind the meaning of search terms as they appear in the document set [2][23]. Many conceptual search methods could be assumed to be semantic approaches too, as they both attempt to understand the meaning behind words as they appear in a document.

One method of semantic search that stands out from conceptual search is the use of ontologies or taxonomies, which help a user attain the formal articulation of a

specific domain [25]. For example a search application for an Electronic Medical Record system could utilize a medical taxonomy to help expand a user's original query to include formal definitions. A user with no formal medical knowledge could find patients complaining of heart-attack like symptoms for a comparative analysis study. They may perform a search *heart-attack* and the taxonomy could help to expand the query to contain a more formal definition like *myocardial infarction*.

### 2.8 Improving Search through Expertise Identification

In [17] the authors use domain experts to improve the relevancy of document search. Although this is not an expertise search engine, rather a traditional enterprise search application, the concepts are still applicable and useful in improving expertise search.

The paper's aim is to improve information retrieval in document search by refining techniques that are solely word-based and creating an expert-based document search method. The idea is to exploit an author's status or reputation in a specific domain when searching within that domain. Essentially the method improves relevancy by boosting documents authored by experts in result rankings.

The methodology used to define expertise is graph-based. The authors use an "expert graph", a directed graph of authors being nodes and directed edges representing an author that has cited another (the source author is the citee and the target author is the cited). Essentially it is a representation of citation relationships.

The “expert graph” is then used to compute an expert reputation score using the equation,

$$\mathrm{E}(u) = \frac{\epsilon}{N} + (1-\epsilon) \cdot \sum_{(u,v)\in G} \frac{E(v)}{DEG\circ(v)}$$

where,

- $\mathrm{E}(u)$ represents author $u$’s reputation.
- $\epsilon$ a damping factor used to prevent 0 scored reputations.
- $N$ is the total number of authors in the graph $G$.
- $DEG \circ (v)$ is the number of citations by author $v$.

Based on the above equation the reputation scores are calculated iteratively. In figure 10 below we have an example “expert graph”. Running the equation for all nodes ranks C and D with the highest reputation scores.

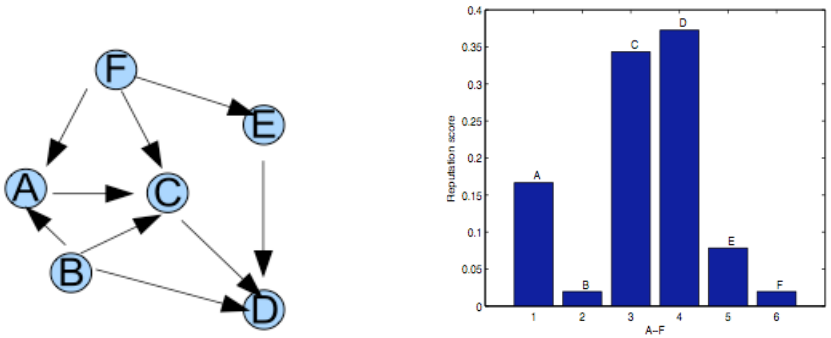



**Figure 10 - expert graph (left) and corresponding reputation scores (right)**

After looking at the example we can see that the equation does a good job at quantitatively assigning an expertise score. In my expertise location system I will make use of an equation to calculate a reputation score. My approach will not require a graph, instead I will be using statistical values to compute an “expertise score” that can be used

to boost expert documents in search result rankings, giving more weight to individuals with greater significance.

In the paper "Designing and Implementation of Expertise Search & Hotspot Detecting System" the authors describe a system to identify experts and hot trending topics [18]. Their system incorporates similar functioning modules that my expertise location system does. One such module relevant to both our systems is the navigation module that provides users an interface to browse and filter instead of search. The navigation module could also be used in conjunction with search, for instance a user first searches for a phrase, retrieves a result set, and then further filters the set based on parametric values. This type of navigation is referred to as faceted search or dynamic navigation [19].

In [18] the author's rely heavily on a thesaurus search module to increase search recall (the number of relevant search hits). In my expertise location system I will also utilize a synonym search module similar to their thesaurus search. Essentially what the module does is query expansion, taking terms of a search phrase and expanding them into synonymous phrases or terms and issuing the additional terms and phrases in the original query.

## 2.9 Ranking Expertise

### *2.9.1 Identifying High Quality Documents*

In [26] Craig Macdonald et al. investigate a new dimension in expertise location by identifying what documents are best for indicating the expertise of a candidate. Techniques designed to predict the quality of a candidate's documents are reviewed.

In Macdonald's work candidates do not have self-profiles or biographies, rather candidates have a document set to represent them. In the enterprise today many organizations require workers to keep a self-authored profile, which is a time killing exercise that many employees find tedious and unproductive.

Macdonald begins his discussion by detailing the importance of selecting on-topic documents in the expertise profiling process. This will improve the precision of the expertise search application, and improve the precision of the expertise candidate profiling. He also mentions the principle that the accumulation of as much information as possible is desired, and this can be aided through query expansion and understanding the data available. His work states the importance of balancing search performance measures; precision and recall.

The aim of Macdonald's research is to find a way to rank a document's importance. He draws a comparison to the goal of website information retrieval, and how it is accomplished through understanding linkages. Macdonald coins a term high-quality expertise evidence to help establish a measurement of importance for an enterprise document. Using an example, he states that a homepage of a candidate should

be given higher importance, than meeting minutes attended by a candidate. Macdonald suggests that a weighting scheme be used to boost a document of greater importance.

In a comparable fashion, I will attempt to gather as much pertinent information into my search index, balance both precision and recall, and calculate a measure of importance for each piece of textual evidence in my expertise location system.

### *2.9.2 The Voting Model*

Expertise search applications require two data components; a list of candidates and a set of textual evidence for each candidate. Most organizations thinking of implementing an expertise locator would need to ensure they have both data requirements met. If profiles for the candidates are not available, then they can be created implicitly through a corpus of documents. Macdonald uses this implicit approach because his dataset does not include explicit profiles for each candidate.

After creating the implicit profiles Macdonald is able to rank them in response to a query using a relevancy calculation based on the Voting Model approach, which considers the problem of expert search as a voting process [27][29][30]. Instead of directly ranking candidates, the method considers a candidate's documents. The process can be described as *R(Q),* where R represents the ranking of a document specific to query *Q*. Every time a document is returned for a query a vote is cast for the document's owner. The ranking of candidates is then modeled by aggregating the votes. Twelve voting techniques are defined in [27], however in Macdonalds research he uses only the expCombMNZ technique because it has proven to be the most robust across all tested

datasets [27]. ExpCombMNZ ranks candidates by the sum of the exponential of the relevance scores of the documents associated to each candidate. The relevance score of a candidate expert *C* with respect to a query *Q*, is,

$$score\_cand(C,Q) = \|R(Q) \cap profile(C)\| \cdot \sum_{d \in R(Q) \cap profile(C)} exp(score(d,Q))$$

where *profile(C)* is the set of documents associated with candidate *C*, and *score(d,Q)* is the relevance score of the document in the document ranking *R(Q)*. □*R(Q)* ∩ *profile(C)*□ is the number of documents from the profile of candidate C that are in the ranking *R(Q)*, and *exp()* is the exponential function. The exponential function boosts candidates that are associated to highly scored documents (strong votes). In my expertise locator I will also use the voting model's summation approach to calculate a score based on the documents returned for each candidate, however my boosting functions I will evaluate will be based on relevancy scores and expertise scores; calculated at index time against the available metadata that defines the quality of a document.

Documents are ranked using the DLH13 document weighting model [28] from the Divergence from Randomness (DFR) framework. DLH13 was chosen because it has no term frequency parameter that requires tuning, as it is inherent to the model. For my application I will use SOLR's scoring model which does consider term frequency, but balances it out by utilizing inverse document frequency, allowing rarer terms to count more than common terms.

The major pitfall of this approach in a real world scenario is that the candidate profiles need to be recalculated on a regular basis because profiles are based off

accumulated works. In a large organization this process would need a considerable amount of processing time and power to keep profiles in synch with all the sources of candidate evidence (emails, fileshares, websites, etc). In my solution I have chosen a streamlined approach that bases expertise off the documents returned by the query, so no implicit profiling is required.

### *2.9.3 Probabilistic Modeling*

In [31] Balog et al. define two probabilistic approaches to identifying experts in a corpus of enterprise content, Model 1 & Model 2. The first of these models expertise by analyzing all the documents a candidate is associated with, and the second approach locates documents on topic and then finds the associated expert. Balog's research reveals a performance advantage using the second approach. In my work I will apply his query based approach, where document's are culled by a query before determining the experts.

Balog's main goal is to understand how the quality of associations between documents and candidates, along with differing search strategies, impact the ability to locate expertise. Document-candidate associations are required for both described approaches. It is assumed that a document *d* is associated with a candidate *ca,* if the document quantifies the candidates expertise or vise-versa. The document-centric perspective is used by Model 1, and can be formalized in terms of probability as,

$$p(d|ca) = \frac{a(d, ca)}{\sum_{d' \in D} a(d', ca)},$$

where *D* is the set of documents. Naturally ranking documents using Model 1 for a candidate *ca* will cause the top documents to be the ones the candidate is strongly associated to.

Model 2 uses a *candidate-centric* approach to calculating the strength of association between documents and candidates. Balog uses the probability,

$$p(ca|d) = \frac{a(d, ca)}{\sum_{ca' \in C} a(d, ca')},$$

where *C* denotes the set of possible candidate experts. The idea behind this probability is that it locates candidates who made the biggest contribution to a specific document.

To summarize Model 1 creates a candidate profile from all their associated documents, and uses this profile to predict their expertise on a certain topic. Instead of creating implicit profiles, Model 2 computes an expertise ranking by first finding the documents related to a topic and then finding the experts based upon this specific result set. In my work I will be adopting an approach similar to Model 2 because it has the advantage of not requiring the need to create and maintain a list of implicitly created candidate profiles.

### 2.10 Evaluating Search Results

Evaluating expertise search is usually performed in the same fashion as enterprise search, via user acceptance testing and performance measures. User

acceptance testing is based on relevancy, and a pass is determined by how well a result set meets the needs of a query. In an expertise search solution the only difference is instead of evaluating documents you evaluate individuals and or groups. User acceptance testing can be difficult to complete due to the uniqueness and size of enterprise content in a search solution. In the industry the task can take weeks testing by domain experts to verify the system is working as desired.

In order to evaluate how well an enterprise application retrieves results relevant to a query, results must be quantified by assigning a relevance score to each result. These scores can be represented in binary (0 entails irrelevant and 1 determines relevance), or graded (indicating results have a degree of match to a query). After scores have been assigned to search results they are sorted and presented to the user in a ranked result set.

Several performance can be used to evaluate the effectiveness of enterprise search. All measures require a document collection and a query or set of queries. Performance measures assume a discrete notion of relevancy; either a document is relevant or it is irrelevant, however in real-life there may be different lights of relevancy. Following is a list of common search performance measures [34],

### *2.10.1 Precision*

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

Precision measures an enterprise search application's ability to retrieve relevant documents, or to be more precise it measures the fraction of retrieved documents that

are relevant. Most testing uses binary relevance judgment where documents are either classified as relevant or irrelevant [35]. However it is possible a document may convey more relevant information than another, and in some studies multiple categories have been used in relevancy judgment [35].

### *2.10.2 Recall*

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

Recall measures the ability for search to bring back all the possible relevant documents on a specific topic or query. It is quite easy to achieve high recall by returning all the documents in the enterprise for any given query, so the recall measure alone cannot be used to determine the effectiveness of an enterprise search solution; precision and recall must be considered in parallel.

### *2.10.3 Fall-out*

$$\text{fall-out} = \frac{|\{\text{non-relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{non-relevant documents}\}|}$$

Fall-out measures the fraction of documents returned by a query that are not relevant. It is desirable to have a low fall-out measure to save users time in culling through information that is not aiding their task.

The figure below is a good visualization for understanding precision, recall, and fallout.

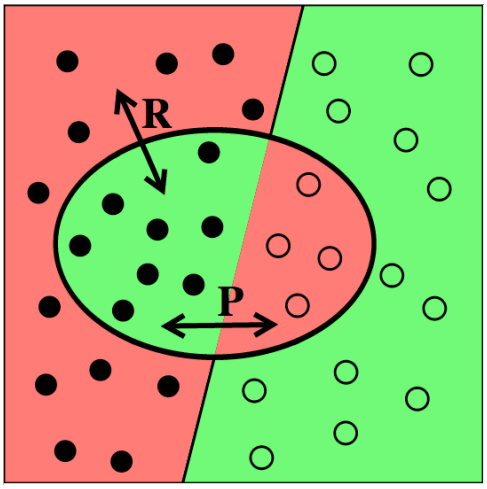



**Figure 11 - performance measure visualization [37]**

Relevant documents are represented by the black dots located on the left side of the square. Retrieved documents are those inside the oval, the left side of the oval (green zone) encapsulates the relevant items retrieved, and the right side of the oval (red zone) represents the fallout.

### *2.10.4 F-Measure*

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

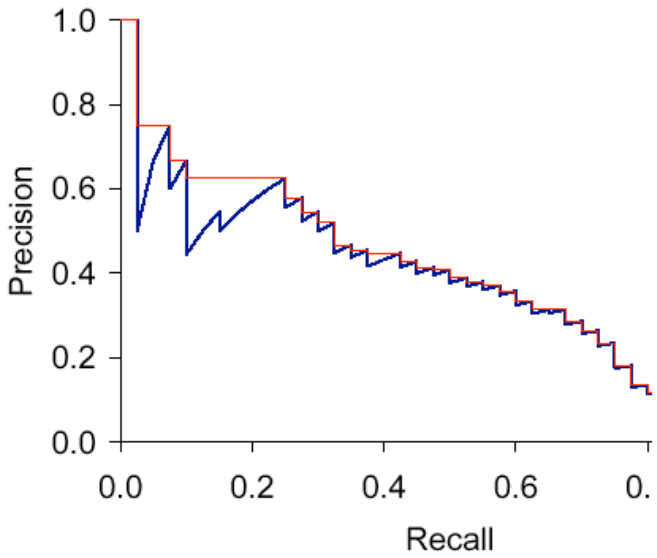
The F-measure combines precision and recall. It is a harmonic mean that evenly weights precision and recall.

### *2.10.5 Average Precision*

$$\text{AveP} = \int_0^1 p(r)dr.$$

In large organizations enterprise content can be vast, and a user query could bring back more results than the user can examine. Therefore, to be desirable a search application should place the most relevant documents at the top of the results. The average precision measure can be used to evaluate ranking of a search application by computing a precision and recall measure at every position in the ranked sequence of documents. The result is a precision-recall curve plotting precision *p(r)* as a function of recall *r*. Average precision is then calculated as the average value of *p(r)* over the interval from *r=0* to *r=1* [36].

### *2.10.6 Precision Recall Curve*




Precision and recall values can be plotted to give a *precision-recall* curve, such as the one shown above. Precision recall curves have a distinctive saw-tooth shape. If the ($k$+1)th document retrieved is irrelevant then recall remains the same for the top $k$ documents, but precision will drop. If the ($k$+1)th document is relevant then both precision and recall increase, and the curve jumps up and to the right. The desired level of precision and recall is subjective, however it is understood that most users would be ok seeing more documents if it increased the number of relevant documents [38].

Looking at the entire precision recall curve can be informative, but in practice this information is boiled down to an 11 point interpolated average precision. The interpolated precision is measured at 0.0, 0.1, 0.2, ..., 1.0. The 11-point interpolated average precision for the graph above is,

| Recall | Interpolated Precision |
|---|---|
| 0.0 | 1.00 |
| 0.1 | 0.67 |
| 0.2 | 0.63 |
| 0.3 | 0.55 |
| 0.4 | 0.45 |
| 0.5 | 0.41 |
| 0.6 | 0.36 |
| 0.7 | 0.29 |
| 0.8 | 0.13 |
| 0.9 | 0.10 |
| 1.0 | 0.08 |

### *2.10.7 Mean Average Precision*

$$\mathrm{MAP} = \frac{\sum_{q=1}^{Q} \mathrm{AveP(q)}}{Q}$$

Ranking for a set of queries can be evaluated using the mean average precision measure by taking the average precision scores for each query, where $Q$ is the number of queries. MAP has shown to be a good measure because of its discrimination and stability [38].

# Chapter Three: **Terrorist Expertise Locator (TEL)**

For my expertise location solution I will be building an expertise search engine for locating expertise among terrorist organizations. My decision to use terrorism related data was made after finding a publicly available data-set of terrorist acts, the Global Terrorism Database (GTD) [39]. The GTD meets the data requirements for creating an expertise location application by having a list of candidates (terrorist group names) and a set of document evidence for each candidate (richly defined terrorist acts pulled from government and news agencies). As far as my research indicates, I have not found any other works that have used this data in an expertise location application. I believe the GTD dataset is a good choice for expertise research because many people can relate to terrorism because it is so prevalent in mainstream media, and thus people can review the results of this research and confirm its effectiveness without requiring a high degree of domain specific knowledge.

I will begin with a discussion of the data, continue on with the intricacies encountered while building the expertise search engine, and conclude with a dialogue on the results.

## 3.1 The Data

I started my search for data with the intent of finding a publicly available export of content from the corporate world. Ideally I was looking for biographies, emails, and documents from a group of individuals. One data set I considered was the Enron email dataset [6]. I decided to forgo this option because I felt email alone would be poor input to determine expertise. I broadened my search and opened myself up to the possibility

of using something outside the corporate realm. Then I came across a relatively new publicly available dataset of terrorist acts, the Global Terrorist Database (GTD) [39]. After investigating the integrity of the data I decided it would suite my needs for a study on expertise search.

There are two data requirements for building an expertise search system; a list of candidates, and some textual evidence of expertise for each candidate [26]. There is a very limited selection of data sets that have been used in this area of research. Most of the authoritative test collections for expertise location research are available from TREC [33], and they have been heavily tested. My choice to use the GTD was made for the following reasons,

- Originality, I wanted to experiment with a new dataset that has not been heavily researched.
- The domain of terrorism knowledge is widely available to the public, and so evaluating the relevancy of results can be performed by cross referencing popular sources available on the web.
- There is a richness of structured and unstructured content for each terrorist act.

I feel that the methodologies used to create my expertise search application could be used in a variety of domains. Specifically, I hope the application could be applied to a medical data set that could be used by general practitioners to locate specialist expertise for patients requiring medical attention.

### *3.1.1 Global Terrorist Database (GTD)*

GTD is the result of several data collection efforts relying on unclassified textual content [39]. Sources include news articles, books, journals, and legal documents. GTD comprises an exhaustive set of terrorist acts that occurred on US soil from 1970 until 2011. The approach used in collection of terrorist acts was to err on the side of inclusiveness, but to also include metadata that can be used to filter out terrorist acts according to the goals and definitions of proposed research.

The GTD defines an act of terrorism as, *the use of illegal force and violence by a non-state actor to attain political, economic, religious, or social goal through fear, coercion, or intimidation*. All three of the following attributes of an attack must be present,

- The attack must be intentional
- The attack must exhibit a level of violence or threat of violence
- The individuals involved must be sub-national actors

In addition to the above attributes two of the following criteria must also be present,

- The act must be aimed at attaining a political, economic, or social goal
- There must be evidence to suggest the act was committed in order to present a message to a larger audience than the victims themselves
- The act must be outside the context of legitimate warfare activities

For further details on the data set please visit the GTD homepage [40].

### *3.1.2 Data Preparation*

The data in the GTD are presented in comma separated value (CSV) format. The GTD table has 104 689 terrorist attacks and 123 columns of data describing each attack.

My first task was to prepare the data for indexing by cleaning it of irrelevant data. I wanted to focus on inclusion of as much unstructured text as possible to mimic the corpus of an enterprise. I removed any columns that did not present content useful for defining expertise. After Culling the dataset I have the following columns,

3.1.2.1 GTD

- eventid - primary key for each entry.
- iyear - year of attack.
- imonth - month of attack.
- iday - day of attack.
- country_txt - country where attack occurred.
- region_txt - region where attack occurred.
- provstate - province where attack occurred.
- city - city where attack occurred.
- location - location where attack occurred
- summary - summary of attack.
- success - was the attack successful.
- suicide - was it a suicide attack.
- attacktype1_txt - attack type (assassination, hijacking, kidnapping, barricade, bombing, unknown, armed assault, unarmed assault).

- targtype1_txt  corp1 - target type (business, government, police, military, abortion, airports/airlines, educational, food/water supply, journalist/media, maritime/ports, NGO, other, private citizens/property, religious facilities, telecommunication, terrorists/militias, tourists, non-aviation transportation, unknown, utilities, violent political parties)
- target1 - specific person, building, etc. that was targeted.
- natlty1_txt - nationality of attacked target.
- gname  - name of terrorist group responsible for attack.
- motive  - motive for attack.
- weaptype1_txt - weapon type used in attack (biological, chemical, radiological, nuclear, firearms, explosives, fake weapons, incendiary, melee, vehicle, sabotage, other, unknown).
- weapsubtype1_txt - more specific information for weapon type above.
- weaptype2_txt - as described in weaptype1_txt.
- weapsubtype2_txt - as described in weapsubtype1_txt.
- weaptype3_txt - as described in weaptype1_txt.
- weapsubtype3_txt - as described in weapsubtype1_txt.
- weapdetail - weapon details.
- nkillus - number of US citizen fatalities.
- nwoundus - number of US citizens injured.
- propvalue - value of property damage.
- propcomment - property damage comment.

- ransompaid - ransom paid.
- ransomnote - details of ransom note.
- hostkidoutcome_txt - kidnapping outcome (attempted rescue, hostage released, hostage escaped, hostage killed, successful rescue, combination, unknown).
- addnotes - additional information that could not be captured in the above fields.
- scite1 - first source used to compile information.
- scite2 - second source used to compile information.
- scite3 - third source used to compile information.

## 3.2 Data Transformation

The data transformation phase takes the raw data and manipulates it in a way to enhance the richness of the content, and hopefully improve relevancy ranking. Tasks performed on the datasets include the manipulation of metadata and the creation of a terrorist attack quality score. Manipulation of metadata is an important task that enables advanced search features such as field boosting, custom ranking, and search [3].

### *3.2.1 Manipulation*

The following tasks were carried out on the GTD,

- All attacks committed by an unknown group were removed
- An attack quality score was computed

### *3.2.2 Attack Quality Score*

An attack quality score was calculated for each terrorist attack in the GTD. The attack quality score is used to improve relevancy ranking of attacks by boosting higher quality attacks. The formula I used to calculate the attack quality score is,

*score_attack(a)* = $s_a + k_a + w_a + p_a + r_a$

where *a* is a terrorist attack and,

$s_a$ = success (0 = not successful attack, and 1 = successful attack),

$k_a$ = normalized value of US citizens killed,

$w_a$ = normalized value of US citizens wounded,

$p_a$ = normalized value of property damage,

$r_a$ = normalized value of ransom paid in kidnapping incidents,

the normalization function used is the standard score equation,

$$\frac{x - \mu}{\sigma}$$

where *x* is the raw value, $\mu$ is the mean, and $\sigma$ is the standard deviation.

## 3.3 Data Ingestion

In order to ingest the data into SOLR a schema must be crafted. The schema tells SOLR about the terrorist accounts in the GTD, specifically what fields the GTD contains, the field's type (string, int, long, etc), and how they should be stored in the

index. How a field is stored is determined by SOLR field attributes. The SOLR field attributes used in my schema include,

- indexed - controls what is searchable.
- stored - controls what is retrievable.
- docValues - improves memory caching for field faceting, grouping, sorting, and function queries.
- multivalued - allows a field to contain multiple values.
- required - makes a field required
- default - assign a default field value if no value is specified.

### *3.3.1 SOLR Schema*

The following SOLR schema defines the field type and attributes for the fields in the GTD that are searchable or retrievable. The metadata fields are all typically assigned stored, docValues, and default attributes. This combination of attributes will allow users to navigate and browse via the metadata values. For example, a user could easily filter results to all successful suicide attacks targeting police. The *eventid* field has unique attributes set (indexed and required) because this field is the unique identifier for the documents in SOLR's search index. This field acts very much like a primary key in a relational database.

The unstructured fields are all assigned the field type *text_en*. This is a special field type that tokenizes, removes stop words, down cases, applies Porter's stemming,

and finally applies synonyms from a synonym text file. The synonym file will be detailed at the end of the field table schema definitions.

**Table 1 - Solr metadata schema**

| Field Name | Field Type | Field Attribute(s) |
|---|---|---|
| eventid | String | indexed, stored, required |
| date | String | stored |
| country_txt | String | stored, docValues, default |
| region_txt | String | stored, docValues, default |
| provstate | String | stored, docValues, default |
| city | string | stored, docValues, default |
| location | string | stored, docValues, default |
| success | boolean | stored |
| suicide | boolean | stored |
| attacktype1_txt | string | stored, docValues, default |
| corp1 | string | stored, docValues, default |
| target1 | string | stored, docValues, default |
| natlty1_txt | string | stored, docValues, default |
| gname | string | stored, docValues, default |
| weaptype1_txt | string | stored, docValues, default |
| weapsubtype1_txt | string | stored, docValues, default |
| weaptype2_txt | string | stored, docValues, default |
| weapsubtype2_txt | string | stored, docValues, default |

| | | |
|---|---|---|
| weaptype3_txt | string | stored, docValues, default |
| weapsubtype3_txt | string | stored, docValues, default |
| nkillus | double | stored, docValues, default |
| nkillusnorm | double | stored, docValues, default |
| propvalue | double | stored, docValues, default |
| propvaluenorm | double | stored, docValues, default |
| ransompaid | double | stored, docValues, default |
| ransompaidnorm | double | stored, docValues, default |
| expscore | double | stored, docValues, default |

**Table 2 - Solr unstructured content schema**

| **Field Name** | **Field Type** | **Field Attribute(s)** |
|---|---|---|
| summary | text_en | stored |
| motive | text_en | stored |
| weapdetail | text_en | stored |
| propcomment | text_en | stored |
| ransomnote | text_en | stored |
| hostkidoutcome | text_en | stored |
| addnotes | text_en | stored |
| scite1 | text_en | stored |
| scite2 | text_en | stored |
| scite3 | text_en | stored |

You may notice none of the fields defined above are indexed, meaning there is nothing searchable defined yet. The searchable fields are constructed by grouping the singular fields defined above via SOLR's *copyField* command. The *copyField* command copies one field to another at the time a document is added to the index. I have used it to index the same field differently, to add multiple fields to the same field, for faster searching, and also to test search against different field configurations. Here are the copy fields I have setup,

**Table 3 - Solr copyFields**

| **Field Name** | **Field Type** | **Field Grouping** |
|---|---|---|
| locationmeta | text_en | country_txt, region_txt, provstate, city, location |
| targetmeta | text_en | targtype1_txt, corp1, target1 |
| attackermeta | text_en | natlty1_txt, gname |
| weaponmeta | text_en | attacktype1_txt, weaptype1_txt, weapsubtype1_txt, weaptype2_txt, weapsubtype2_txt, weaptype3_txt, weapsubtype3_txt |
| metadata | text_en | all metadata fields |
| content | text_en | motive, summary, ransomnote, hostkidoutcome_txt, addnotes, scite1, scite2, scite3 |

### 3.4 Data Analysis

Analysis of data is the process of inspecting and modeling the content with the goal of highlighting useful information and supporting decision making [3]. Data

analysis has multiple facets and approaches. In my analysis I will be looking at the most commonly occurring terms in the indexed fields. The field analysis will help me to locate important terms to consider for synonym expansion. I will also use the results of my field analysis to construct my test queries, which I will use to evaluate the relevancy and ranking of the system. I have decided to perform the field analysis after tokenizing, stripping stop words,  and down casing words. This will prevent punctuation and case sensitivity from effecting the term frequency statistics used to find the top commonly occurring words.

### *3.4.1 Stopwords*

In enterprise search, stopwords are words removed from the search index during the ingestion phase. There is not one definite list of stopwords which all tools use, however search technologies are typically shipped with a default set of stopwords for popular languages.

Any group of words can be chosen as stop words for a given purpose. Default stop word lists contain common, short function words, such as *the, is, at, which*, and *on*. In this case, stop words can cause problems when searching for phrases that include them, particularly in names such as 'The Who', 'The The', or 'Take That', and this is why stop word lists can be configured for a search engine's purpose and the data it operates on. Adding common words to a stopword list also benefits search time performance, as the index is trimmed of all stopwords.

Here are the stopwords used in my search engine,

- a
- an
- and
- are
- as
- at
- be
- but
- by
- for
- if
- in
- into
- is
- it
- no
- not
- of
- on
- or
- such
- that
- the
- their
- then
- there
- these
- they
- this
- to
- was
- with

### *3.4.2 Term Analysis for Location Metadata*

Here are the top 50 terms after tokenizing, removing stopwords, and down casing the location grouping of metadata,

**Table 4 - top 50 location terms**

| | |
|---|---|
| 25,016 | south |
| 21,693 | america |
| 14,793 | asia |
| 11,956 | africa |
| 11,171 | north |
| 11,087 | europ |
| 11,019 | western |
| 8,778 | east |
| 8,238 | middl |
| 7,119 | central |
| 6,852 | caribbean |
| 5,670 | attack |
| 5,490 | colombia |
| 5,162 | peru |
| 4,892 | place |
| 4,856 | took |
| 4,583 | el |
| 4,538 | india |
| 3,895 | provinc |
| 3,872 | salvador |
| 3,853 | northern |
| 3,791 | sub |
| 3,747 | saharan |
| 3,646 | ireland |
| 3,490 | southeast |
| 3,459 | state |
| 2,778 | district |
| 2,777 | san |
| 2,629 | spain |
| 2,460 | unknown |
| 2,407 | sri |
| 2,406 | lanka |
| 2,291 | philippin |
| 2,106 | west |
| 1,850 | lima |
| 1,841 | unit |
| 1,840 | occur |
| 1,793 | afghanistan |
| 1,776 | belfast |
| 1,770 | nicaragua |
| 1,755 | u. |
| 1,739 | villag |
| 1,658 | turkei |
| 1,608 | near |
| 1,585 | citi |
| 1,519 | algeria |
| 1,363 | pakistan |
| 1,356 | area |
| 1,343 | chile |
| 1,157 | depart |

### *3.4.3 Term Analysis for Target Metadata*

Here are the top 50 terms after tokenizing, removing stopwords, and down casing the target grouping of metadata,

**Table 5 - top 50 target terms**

| | |
|---|---|
| 11,750 | private |
| 11,655 | citizens |
| 11,623 | property |
| 9,245 | government |
| 9,148 | military |
| 9,029 | business |
| 8,033 | police |
| 7,541 | general |
| 5,422 | targeted |
| 3,938 | unit |
| 3,303 | utilities |
| 3,139 | govt |
| 2,960 | transportation |
| 2,803 | civilians |
| 2,051 | office |
| 2,000 | attack |
| 1,948 | were |
| 1,887 | station |
| 1,854 | line |
| 1,748 | unknown |
| 1,715 | party |
| 1,682 | army |
| 1,667 | bank |
| 1,581 | bus |
| 1,549 | tower |
| 1,545 | high |
| 1,524 | town |
| 1,367 | tension |
| 1,350 | patrol |
| 1,341 | vehicle |
| 1,335 | diplomatic |
| 1,315 | village |
| 1,294 | unk |
| 1,272 | institution |
| 1,235 | educational |
| 1,174 | post |
| 1,161 | national |
| 1,100 | religious |
| 1,085 | institutions |
| 1,078 | figures |
| 992 | journalists |
| 991 | media |
| 971 | co |
| 941 | company |
| 904 | catholic |
| 898 | other |
| 886 | power |
| | israeli |
| 874 | electrical |
| 834 | civilian |

### *3.4.4 Term Analysis for Weapon Metadata*

Here are the top 50 terms after tokenizing, removing stopwords, and down casing the weapon grouping of metadata,

**Table 6 - top 50 weapon terms**

| | |
|---|---|
| 32,028 | unknown |
| 26,240 | type |
| 25,871 | bombs |
| 25,828 | explosives |
| 25,788 | dynamite |
| 24,455 | firearms |
| 24,309 | bombing |
| 24,280 | explosion |
| 19,131 | explosive |
| 17,280 | assault |
| 17,029 | armed |
| 13,707 | firearm |
| 13,025 | automatic |
| 12,643 | weapon |
| 10,214 | attack |
| 8,147 | assassination |
| 8,097 | gun |
| 7,064 | used |
| 5,387 | incendiary |
| 4,874 | were |
| 4,359 | facility |
| 4,358 | infrastructure |
| 4,014 | taking hostage |
| 3,789 | handgun |
| 3,424 | kidnapping |
| 3,260 | pistol |
| 2,783 | other |
| 2,361 | bomb |
| 2,010 | mortars |
| 2,004 | device |
| 2,003 | rockets |
| 1,999 | projectile |
| 1,997 | etc rpgs |
| 1,787 | melee |
| 1,590 | object |
| 1,560 | weapons |
| 1,559 | grenade |
| 1,509 | fire |
| 1,365 | arson |
| 1,329 | improvised |
| 1,211 | knife |
| 1,209 | sharp |
| 1,152 | mine |
| 1,135 | land |
| 971 | vehicle |
| 912 | incident |
| 759 | gasoline |
| 720 | tnt |

### *3.4.5 Term Analysis for Unstructured Content*

Here are the top 20 terms after tokenizing, removing stopwords, and down casing the unstructured content,

**Table 7 - top 50 unstructured content terms**

| | |
|---|---|
| 13,820 | unknown |
| 12,477 | attack |
| 10,502 | were |
| 9,001 | responsibility |
| 8,435 | killed |
| 8,432 | claimed |
| 7,192 | group |
| 7,159 | motive |
| 6,998 | specific |
| 6,868 | ó |
| 6,725 | two |
| 6,692 | one |
| 6,488 | suspected |
| 5,682 | news |
| 4,510 | police |
| 4,390 | press |
| 4,217 | militants |
| 4,113 | three |
| 4,051 | http |
| 4,042 | from |
| 3,970 | national |
| 3,956 | presse |
| 3,928 | bomb |
| 3,909 | attacks |
| 3,857 | injured |
| 3,701 | casualties |
| 3,669 | france |
| 3,659 | reported |
| 3,593 | incident |
| 3,592 | agence |
| 3,352 | incidents |
| 3,294 | 11 |
| 3,279 | members |
| 3,230 | terrorism |
| 3,198 | 10 |
| 3,105 | perpetrators |
| 3,097 | wounded |
| 2,990 | 12 |
| 2,985 | lexisnexis |
| 2,975 | academic |
| 2,955 | people |
| 2,883 | responsible |
| 2,855 | assailants |
| 2,839 | near |
| 2,833 | kill |
| 2,827 | center |
| 2,821 | india |
| 2,772 | rebels |
| 2,771 | four |
| 2,715 | 2010 |

### *3.4.6 Synonyms*

Synonyms for my expertise search have been designed to broaden descriptive terms and skills for the terrorist organizations. I have chosen to use equivalent synonyms, which means if any of the possible synonyms are hit in a user query it will expand the query to include all other terms defined as synonyms.

Using the term analysis above I found interesting terms that are synonymous and combine them. Afterwards I expanded terms further by referencing a thesaurus [41]. The synonyms of interest to me are skill related because the nature of an expertise search solution is to return the top ranked experts for a given task. Because of this most of the terms defined in my synonym file are of weapons and targets, as these are the key entities in a terrorist attack.

#### 3.4.6.1 Target Synonyms

- america, united states, united states of america, usa, us
- government, govt, regime, agency
- police, authority, force, officer
- business, office, company, corporation, headquarters, hq
- vehicle, truck, car, bus, motorcycle, automobile
- building, facility, tower, property, skyscraper, high-rise
- military, army, force, militia
- press, presse
- university, college, school

3.4.6.2 Skill Synonyms

- bomb, explosive, dynamite, explosion, incendiary, tnt, dynamite, grenade, mine, device
- missile, rpg, rocket
- gun, firearm, pistol, revolver, rifle, shotgun, carbine, automatic, handgun, semiautomatic, mortar
- kidnap, abduct, capture, hostage
- kill, murder, execute, assassinate, casualties, casualty
- destroy, demolish, knock down, blast, blow up, explode, wreck, ruin, raze, damage, disable, cripple
- injure, maim, wound, incapacitate, hurt, harm
- phone, cell, mobile, cellular, pager, telephone
- fire, arson, blaze
- fight, melee, skirmish

### *3.4.7 Index Time Synonym Expansion VS Query Time Synonym Expansion.*

There are two options available when applying synonyms in SOLR, and they are to apply synonyms at query time or at index time. If you decide to apply synonyms at query time, as I did, your query will be analyzed for any possible terms in your synonym file, and if any are found your query will be expanded to include the synonymous terms. If applying synonyms at index time, your index will be expanded to include synonymous terms when any term is located in the synonym file during indexing.

The intuitive choice may be to go with query time expansion. This choice has the benefits of not growing your index size and allowing for dynamic synonyms because you do not have to re-index to leverage changes to your synonym file. However there are drawbacks; multi-word synonyms won't work as phrase queries, rare synonyms will be boosted causing unintuitive results, and multi-word synonyms won't be matched in queries.

The reason multi-word synonyms wont be matched in a phrase query (a phrase query is searching for an exact match of a specific phrase, usually encapsulated in double quotes) is because the query time synonym expansion transforms the user's original query, and the phrase no longer exists as it was entered by the user.

Rare synonyms will be boosted, is also a potential issue of query time expansion. Given the synonym "cat, pussy, kitten", if a user searches for *cat* they will probably get back search results at the top that do not have *cat*, but rather *pussy* or *kitten*. This is because the inverse document frequency (IDF) scoring model gives more weight to term hits that are less frequently occurring in the document collection.

The last major drawback found with query time expansion of synonyms is that multi-word synonyms won't be matched in queries. This is because the tokenizer breaks up input before the synonym analyzer can transform it. For example, consider the synonym "america, united states, united states of america, usa, us ", and the query "terrorists in the united states". The query will be tokenized to [ "terrorists", "in", "the", "united", "states" ], and so no synonym expansion will take place.

Looking at index time expansion, the most concerning issues that arise are that the size of the index will grow, and making changes to the synonym file requires re-indexing of the document collection.

## 3.5 Performance Measure Testing

In order to test performance measures I need to have a set of queries to run against my search engine, and a set of system configurations to test them against. My queries should be formed from terms describing a particular skill or task that is common among terrorist groups and prevalent in the dataset. I will design my queries to include high frequency terms and synonyms, to ensure I have enough results to effectively evaluate performance measures, synonym functionality, and calculate experts. All queries will be tested against the system under specific settings in order to examine the effectiveness of relevancy tuning enhancements made throughout the systems build up. Following is a list of the queries I have constructed for evaluation purposes,

### *3.5.1 Test Queries*

- explosives
- suicide bombing
- ransom kidnapping
- police killing
- government assassination
- sniper attack

### *3.5.2 System Configuration and Tuning*

Using various SOLR system configuration settings I will try and find the ideal setup for my terrorist data set. I will analyze performance measures as I progress through the development of the search engine. As a starting point I will configure SOLR as a basic keyword search engine. At each stage of development I will post precision and recall performance measures. My precision measure will be evaluated for precision in the top 50 documents (I will cast a binary judgment of relevancy on the top 50 documents returned by each query). My recall measure will evaluate to the number of returned documents for each query. I have simplified the recall measure because of time constraints (I do not have the time or resources to assign judgment on all 60139 documents in the test collection). At the end of this work a configuration set will be chosen as the basis for developing the expertise search system.

### *3.5.3 Keyword Search*

The keyword search configuration set will produce a searchable index of terms that are split on whitespace only. All tokens will then be down cased to allow for case-insensitive matching of terms. This type of configuration represents basic search functionality, similar to the search available in basic text editors.

**Table 8 - keyword search recall**

| **Query** | **Results** |
|---|---|
| explosives | 510 |
| suicide bombing | 1829 |

| | |
|---|---|
| ransom kidnapping | 469 |
| police killing | 5756 |
| government assassination | 1858 |
| sniper attack | 10561 |

After reviewing the results of the initial queries it is apparent that "OR" logic is being applied between terms, and the number of results returned for multi-term queries is very high as a result. For example, the query *"sniper attack"* returns 10561 documents out of a total 60139 documents, that is 17.6% of the total documents. I can say with a high degree of certainty that most of the returned documents only contain the word *"attack"*, and not the term *"sniper"*, making the majority of the returned documents irrelevant to the query.

### *3.5.4 Boolean Query Logic*

To cull the irrelevant documents I can change the default search behaviour from the logical OR to the logical AND operator. Here are the results using the logical "AND" operator,

**Table 9 - comparing recall; Boolean AND vs. Boolean OR**

| **Query** | **Results - AND Logic** | **Results - OR Logic** |
|---|---|---|
| explosives | 510 | 510 |
| suicide bombing | 226 | 1829 |
| ransom kidnapping | 35 | 469 |
| police killing | 563 | 5756 |

| | | |
|---|---|---|
| government assassination | 14 | 1858 |
| sniper attack | 9 | 10561 |

Using the AND operator brought search results down dramatically for the multi-term test queries. Now we have fewer results returned for those multi-term queries. In particular, the multi-term queries *"ransom kidnapping"*, *"government assassination"*, and *"sniper attack"*. From this point forward all queries with multiple terms will be tied together with the logical AND operator to tighten up precision.

**Table 10 - keyword search precision @ 50**

| **Query** | **Precision@50** |
|---|---|
| explosives | 1.0 |
| suicide bombing | 1.0 |
| ransom kidnapping | 1.0 |
| police killing | 1.0 |
| government assassination | 0.78 |
| sniper attack | 0.88 |

Precision measures are very high in this keyword search configuration using Boolean AND logic between search terms. This style of search is generating highly relevant documents throughout the result set. That being said, I suspect there are many relevant documents that are not being retrieved because of the tight constraints needed to trigger a document as a hit using this configuration set. For example, the first query *"explosives"* only brings back documents that

contain the term *"explosives",* it will not return a document containing close variants like *"explosive"* or *"explosion"*.

### *3.5.5 Advanced Search*

In order to return documents in the test collection that contain close variants to the query terms we need to implement additional term manipulation features. For the advanced search configuration I will be adding porter's stemming algorithm, a standard tokenizer which lowercases characters and drops non-letters, and an English possessive filter that removes *" 's "* from the end of any terms. These term manipulators will be applied on the index as well as the query terms. Adding these features should increase system recall, and get closer to finding a balance between precision and recall.

**Table 11 - advanced search recall**

| **Query** | **Results** | **Improvement** |
|---|---|---|
| explosives | 3693 | +624% |
| suicide bombing | 636 | +181% |
| ransom kidnapping | 149 | +326% |
| police killing | 2729 | +385% |
| government assassination | 110 | +686% |
| sniper attack | 28 | +211% |

As expected the recall has significantly increased for all queries. The ones that experienced the highest increase in recall were *"explosives"* and *"government assassination"*. These two queries have the greatest array of stems pertinent to the content indexed, and that is why they experienced the highest recall gain. In the next part of testing I will look at how much the advanced settings impacted precision.

**Table 12 - advanced search precision @ 50**

| **Query** | **Precision@50** | **Degradation** |
|---|---|---|
| explosives | 1.0 | 0% |
| suicide bombing | 0.96 | -4% |
| ransom kidnapping | 1.0 | 0% |
| police killing | 0.94 | -6% |
| government assassination | 0.74 | -5% |
| sniper attack | 0.86 | -2% |

Precision did drop, but only slightly. I believe we still have potential to increase recall at a minimal cost to precision by expanding our query to include the synonyms I defined previously.

### *3.5.6 Synonym Search*

Expanding queries on synonymous terms should prove beneficial to retrieving relevant documents that do not contain exact term matches or close variants of a query's terms. I will

implement query side synonym functionality, where synonyms are applied at query time. I have chosen to apply synonyms at query time because the advantage of doing it at index time is to gain the ability to expand multi-word synonyms, but there are no multi-word synonyms present in any of my test queries, so the benefit is not realized. Synonyms will be applied after tokenization but before stemming, in order to match the synonyms I have defined. Now I will examine the change in recall over the previous Advanced Search configuration.

**Table 13 - synonym search recall**

| **Query** | **Results** | **Improvement** |
|---|---|---|
| explosives | 7784 | +111% |
| suicide bombing | 855 | +34% |
| ransom kidnapping | 172 | +15% |
| police killing | 6410 | +135% |
| government assassination | 5320 | +4736% |
| sniper attack | 28 | 0% |

### *3.5.7 Expertise Search*

Before calculating expertise I have decided to calculate a fall-out measure to establish at what point search results begin to degrade; include irrelevant documents. This measure can help us to develop a rank based scoring measure, and determine a cut-off point in the result set , allowing for as few irrelevant documents to be included in the expertise calculation. The traditional fall-out measure requires all documents in the collection to be categorized or labelled

to evaluate it. The level of effort and time required to make such an assessment is out of scope for this work, so I will improvise and use a fall-out measure that is evaluated by determining at what point in the ranking irrelevant documents begin to appear. I will mark the rank of the first three irrelevant documents, the "-" character signifies no irrelevant document was found.

**Table 14 - synonym search fall-out**

| **Query** | **1st Irrelevant Doc Rank** | **2nd Irrelevant Doc Rank** | **3rd Irrelevant Doc Rank** | **Total Results** |
|---|---|---|---|---|
| explosives | - | - | - | 7784 |
| suicide bombing | 51 | 57 | 104 | 855 |
| ransom kidnapping | - | - | - | 172 |
| police killing | 103 | 156 | 198 | 6410 |
| government assassination | 83 | 167 | 176 | 5320 |
| sniper attack | 7 | - | - | 28 |

I have noticed that queries that include single terms and terms that are related have the lowest fallout. For example there were no irrelevant results found in the queries "explosives" and "ransom kidnapping". On the flipside multi-term queries that contain unrelated terms do have a recordable fall-out, such as "suicide bombing", "police killing, and "government assassination".

The results indicate that the ranking of search results could be factored into the calculation of expertise to ensure irrelevant events do not have as much impact.

Chapter Four: **Determining Expertise Among Terrorist Groups**

I will evaluate 3 expertise calculation methods to rank terrorist groups,

- Count-based by frequency in result set.
- Relevancy weighted by relevancy score value of a terrorist attack.
- Expertise weighted by terrorist attack expertise score value.

Each method combines unique weighting and ranking to provide the top 10 experts for each test query. Important ranking differences that are discussed in detail are highlighted in the result tables.

### 4.1 Count-Based

The count-based method determines experts by their frequency in a query's result set. Each terrorist event counts toward the overall score for that terrorist group. The terrorist groups are then sorted by event count to rank the top experts for a user's query.

$\sum_{i=1}^{n} x_i$ = Expertise score for terrorist group $\chi$, where $x_i$ is a terrorist event associated to group $\chi$ in the search result set. Events $x_1$ to $x_n$ cover all events for group $\chi$, in response to a user query.

Each event counts as a single point towards the group's expertise score.

### 4.2 Relevancy Weighted

The relevancy weighted method factors in the relevancy score of each terrorist event when determining expertise. The relevancy score is provided by SOLR's term frequency inverse document frequency (TF-IDF) scoring model [83], in response to a user query.

$\sum_{i=1}^{n} x_i \cdot r$ = Expertise score for terrorist group $\chi$, where $x_i$ is a terrorist event in the search result set. Each terrorist event counts as a single point towards the final score, and is factored by $r$, the relevancy score assigned by Solr's TF-IDF scoring model.

### 4.3 Expertise Weighted

The expertise weighted method factors in the previously defined attack quality score for a terrorist event.

$\sum_{i=1}^{n} x_i \cdot s$ = Expertise score for terrorist group $\chi$, where $x_i$ is a terrorist event in the search result set. Each terrorist event counts as a single point towards the final score, and is factored by $s$, the attack quality score.

### 4.4 Results, Observations, and Validation

To validate the rankings of TEL I have cross referenced unindexed sources, from popular news outlets and government agency reports, to see if their content substantiates my system's findings. I will focus validation on physical characteristics such as group size, area of influence, how long they have been active, and also corroborate their skills against the details presented in their most notorious attacks.

### *4.4.1 "Explosives" Query*

**Table 15 - count-based ranking for query "explosives"**

| Rank | Terrorist Group | Count |
|---|---|---|
| *1 | Taliban | 785 |
| 2 | Revolutionary Armed Forces of Colombia (FARC) | 475 |
| 3 | Liberation Tigers of Tamil Eelam (LTTE) | 369 |
| 4 | Communist Party of India - Maoist (CPI-M) | 353 |
| 5 | Algerian Islamic Extremists | 226 |
| 6 | Basque Fatherland and Freedom (ETA) | 221 |
| 7 | Chechen Rebels | 196 |
| 8 | Moro Islamic Liberation Front (MILF) | 149 |
| 9 | United Liberation Front of Assam (ULFA) | 142 |
| 9 | Islamic State of Iraq (ISI) | 142 |

**Table 16 - relevancy weighted ranking for query "explosives"**

| Rank | Terrorist Group | Score |
|---|---|---|
| *1 | Taliban | 197.7451542 |
| 2 | Revolutionary Armed Forces of Colombia (FARC) | 132.8011749 |
| 3 | Liberation Tigers of Tamil Eelam (LTTE) | 92.6205748 |
| 4 | Communist Party of India - Maoist (CPI-M) | 72.34310495 |
| 5 | Basque Fatherland and Freedom (ETA) | 67.78078652 |
| 6 | Algerian Islamic Extremists | 64.90570062 |
| 7 | Chechen Rebels | 63.061819 |
| 8 | Moro Islamic Liberation Front (MILF) | 41.57087157 |
| 9 | Islamic State of Iraq (ISI) | 38.58911263 |
| 10 | United Liberation Front of Assam (ULFA) | 38.29696698 |

**Table 17 - expertise weighted ranking for query "explosives"**

| Rank | Terrorist Group | Score |
|---|---|---|
| *1 | Taliban | 578.2153118 |
| 2 | Revolutionary Armed Forces of Colombia (FARC) | 292.3606966 |
| 3 | Liberation Tigers of Tamil Eelam (LTTE) | 259.4566006 |
| 4 | Communist Party of India - Maoist (CPI-M) | 256.5306009 |
| 5 | Basque Fatherland and Freedom (ETA) | 145.1253873 |
| 6 | Algerian Islamic Extremists | 139.042218 |
| 7 | Chechen Rebels | 115.5410695 |
| 8 | Islamic State of Iraq (ISI) | 108.7546425 |
| 9 | United Liberation Front of Assam (ULFA) | 106.2389335 |
| **10 | Al-Qa`ida in Iraq | 104.5808171 |

*The top ranked terrorist group is unanimously Taliban. The group shows a peak level of membership in 2001 of ~45,000 individuals [51]. In comparison the second ranked group FARC has an estimated membership of 8,000-10,000 [52], and the third ranked group LTTE had 11,664 members before surrendering to the Sri Lankan military [53]. The membership numbers of these groups and their relative expertise scores, from all three methods of expertise calculation, have a strong correlation.

The Taliban are well known for their use of improvised explosive devices [54][55]. It makes perfect sense that the group is listed as number one for explosives since they have been actively at war with the US and allied forces in their home territory of Afghanistan for the last 10 years [56][57][58].

**The only significant difference between the results of the three expertise ranking methods is the appearance of group Al-Qa'ida in Iraq in the expertise weighted method. I expected Al-Qa'ida to rank higher, as they are prominently featured in mainstream news for their

orchestration of the 1993 World Trade Center Bombing [59] and 2001 September 11th attacks [60]. I believe there are a couple reasons why they did not rank higher in my system,

1. My system favors frequency of attacks over quality of attacks, and Al-Qa'ida are known for their quality of attacks, and not as much for their frequency of attacks.
2. The dataset does not group together attacks from all Al-Qa'ida factions. For example, there are groups dedicated to Al-Qa'ida in Iraq, Al-Qa`ida in the Arabian Peninsula, Al-Qa`ida in the Lands of the Islamic Maghreb, Al-Qa'ida in Lebanon, Al-Qa'ida in Saudi Arabia, and Al-Qa'ida Network for Southwestern Khulna Division.

That being said, it is interesting that the expertise method of calculation brings a faction of Al-Qa'ida into the top 10. I would say this is because the expertise method gives more weight to the quality of attacks versus the simple count-based method, and relevancy weighted method.

### *4.4.2 "Suicide Bombing" Query*

**Table 18 - count-based ranking for query "suicide bombing"**

| Rank | Terrorist Group | Count |
|---|---|---|
| 1 | Taliban | 192 |
| 2 | Al-Qa`ida in Iraq | 73 |
| 3 | Liberation Tigers of Tamil Eelam (LTTE) | 72 |
| 4 | Tehrik-i-Taliban Pakistan (TTP) | 49 |
| 5 | Hamas (Islamic Resistance Movement) | 48 |
| 6 | Islamic State of Iraq (ISI) | 34 |
| 7 | Al-Aqsa Martyrs Brigade | 33 |
| 8 | Palestinian Islamic Jihad (PIJ) | 27 |
| 9 | Al-Qa`ida in the Lands of the Islamic Maghreb (AQLIM) | 26 |
| **10 | Chechen Rebels | 25 |

**Table 19 - relevancy weighted ranking for query "suicide bombing"**

| Rank | Terrorist Group | Score |
|---|---|---|
| 1 | Taliban | 71.6303576 |
| 2 | Liberation Tigers of Tamil Eelam (LTTE) | 26.02623276 |
| 3 | Al-Qa`ida in Iraq | 25.91129969 |
| 4 | Hamas (Islamic Resistance Movement) | 18.50443199 |
| 5 | Tehrik-i-Taliban Pakistan (TTP) | 16.32768502 |
| 6 | Al-Aqsa Martyrs Brigade | 13.54767676 |
| 7 | Islamic State of Iraq (ISI) | 13.28047649 |
| **8 | Chechen Rebels | 12.24385567 |
| 9 | Palestinian Islamic Jihad (PIJ) | 10.75772388 |
| 10 | Al-Qa`ida in the Lands of the Islamic Maghreb (AQLIM) | 10.72550083 |

**Table 20 - expertise weighted ranking for query "suicide bombing"**

| Rank | Terrorist Group | Score |
|---|---|---|
| 1 | Taliban | 135.3961764 |
| 2 | Al-Qa`ida in Iraq | 60.16482996 |
| 3 | Liberation Tigers of Tamil Eelam (LTTE) | 53.79487937 |
| 4 | Hamas (Islamic Resistance Movement) | 36.68960172 |
| 5 | Tehrik-i-Taliban Pakistan (TTP) | 36.38526759 |
| *6 | Haqqani Network | 26.18625688 |
| 7 | Islamic State of Iraq (ISI) | 25.15037924 |
| *8 | Ansar al-Sunna | 22.87919781 |
| 9 | Al-Aqsa Martyrs Brigade | 20.85130239 |
| *10 | Al-Qa`ida | 19.44721989 |

Again, Taliban is a clear leader in all three methods. I previously analyzed membership level as a contributing factor, and I believe these results further substantiate that terrorist group membership has a strong correlation to expertise in my system.

*The expertise weighted method presents the most differing results from the other two methods, with three new groups appearing; Haqqani Network, Ansar al-Sunna, and Al-Qa'ida. This indicates that the quality of their attacks is significant enough to give them a boost in

rankings. The Haqqani Network is known for their use of asymmetric warfare, which includes suicide bombing, and is known to have pioneered the use of suicide attacks in Afghanistan [61]. They are also allied with the Taliban [61], who ranked number one for suicide attacks in this test case.

**One group who fell out of the top 10 in the expertise method was the Chechen Rebels. Cross referencing of unindexed material suggests that this group only recently started using this tactic [62][63][64]. The Chechen suicide bombers are also unique in that they perform the act not out of religious motivation, like the other groups ranked higher, but out of sheer rage against the murderers of their significant others [62]. The difference in motivation and later adoption of suicide bombing provide evidence that their expertise is lesser to groups that have been practicing the tactic religiously for a longer period of time.

### *4.4.3 "Ransom Kidnapping" Query*

**Table 21 - count-based ranking for query "ransom kidnapping"**

| Rank | Terrorist Group | Count |
|---|---|---|
| *1 | Revolutionary Armed Forces of Colombia (FARC) | 31 |
| *2 | Abu Sayyaf Group (ASG) | 24 |
| 3 | Salafist Group for Preaching and Fighting (GSPC) | 17 |
| 4 | Taliban | 12 |
| 5 | National Liberation Army of Colombia (ELN) | 9 |
| 6 | Moro Islamic Liberation Front (MILF) | 5 |
| 7 | National Liberation Front of Tripura (NLFT) | 4 |
| 7 | Dima Halao Daoga (DHD) | 4 |
| 8 | Al-Qa`ida in the Lands of the Islamic Maghreb (AQLIM) | 3 |
| 8 | Communist Party of India - Maoist (CPI-M) | 3 |

**Table 22 - relevancy weighted ranking for query "ransom kidnapping"**

| Rank | Terrorist Group | Score |
|---|---|---|
| *1 | Revolutionary Armed Forces of Colombia (FARC) | 29.99586161 |
| *2 | Abu Sayyaf Group (ASG) | 21.75307061 |
| 3 | Salafist Group for Preaching and Fighting (GSPC) | 15.1840189 |
| 4 | Taliban | 10.71559044 |
| 5 | National Liberation Army of Colombia (ELN) | 8.8838227 |
| 6 | Moro Islamic Liberation Front (MILF) | 4.55440333 |
| 7 | National Liberation Front of Tripura (NLFT) | 3.62157009 |
| 8 | Dima Halao Daoga (DHD) | 3.03783824 |
| 9 | Al-Qa`ida in the Lands of the Islamic Maghreb (AQLIM) | 2.6439376 |
| 10 | Communist Party of India - Maoist (CPI-M) | 2.3107955 |

**Table 23 - expertise weighted ranking for query "ransom kidnapping"**

| Rank | Terrorist Group | Score |
|---|---|---|
| *1 | Revolutionary Armed Forces of Colombia (FARC) | 24.48658942 |
| *2 | Abu Sayyaf Group (ASG) | 18.93324153 |
| 3 | Salafist Group for Preaching and Fighting (GSPC) | 16.6626994 |
| 4 | Taliban | 9.400400386 |
| 5 | National Liberation Army of Colombia (ELN) | 7.068054076 |
| 6 | Moro Islamic Liberation Front (MILF) | 3.91679073 |
| 7 | National Liberation Front of Tripura (NLFT) | 3.133933184 |
| 8 | Dima Halao Daoga (DHD) | 3.133344612 |
| 9 | Al-Qa`ida in the Lands of the Islamic Maghreb (AQLIM) | 2.350081769 |
| 10 | Communist Party of India - Maoist (CPI-M) | 2.350081769 |

This test query on ransom kidnapping is the only one where rankings are unchanged across all three expertise calculation methods. *The top ranked group is FARC, and cross-referencing indicates that their operations are heavily funded by kidnap to ransom activities [65]. FARC has also been operative since 1964 [66], longer than any other ranked group.

*The second ranked group, Abu Sayyaf Group, has been carrying out ransom kidnappings in the Philippines since 1990 [67]. The group's motive for kidnapping is financially

driven [68]. One of their more notorious kidnappings included taking over 20 hostages from a Malaysian dive resort in 2000 [69].

The results of this expertise ranking correlate top ranked groups to financial dependence on ransom kidnappings. Larger more prevalent groups, like the Taliban, rank lower because they draw more funds from other activities like illicit drug manufacturing [70].

### *4.4.4 "Police Killing" Query*

**Table 24 - count-based ranking for query "police killing"**

| Rank | Terrorist Group | Score |
|---|---|---|
| *1 | Communist Party of India - Maoist (CPI-M) | 755 |
| *2 | Taliban | 708 |
| 3 | Revolutionary Armed Forces of Colombia (FARC) | 433 |
| 4 | Liberation Tigers of Tamil Eelam (LTTE) | 248 |
| 5 | Algerian Islamic Extremists | 146 |
| 6 | New People's Army (NPA) | 125 |
| 7 | Chechen Rebels | 121 |
| 8 | Salafist Group for Preaching and Fighting (GSPC) | 98 |
| 9 | Maoists | 94 |
| 10 | Al-Qa`ida in Iraq | 90 |

**Table 25 - relevancy weighted ranking for query "police killing"**

| Rank | Terrorist Group | Score |
|---|---|---|
| *1 | Taliban | 230.1832114 |
| *2 | Communist Party of India - Maoist (CPI-M) | 213.1566371 |
| 3 | Revolutionary Armed Forces of Colombia (FARC) | 138.7017999 |
| 4 | Liberation Tigers of Tamil Eelam (LTTE) | 72.55553585 |
| 5 | Algerian Islamic Extremists | 53.5945741 |
| 6 | New People's Army (NPA) | 36.98957463 |
| 7 | Chechen Rebels | 36.81139771 |
| 8 | Salafist Group for Preaching and Fighting (GSPC) | 36.44211354 |
| 9 | Maoists | 32.52094292 |
| 10 | Al-Qa`ida in Iraq | 28.87878877 |

**Table 26 - expertise weighted ranking for query "police killing"**

| Rank | Terrorist Group | Score |
|---|---|---|
| *1 | Communist Party of India - Maoist (CPI-M) | 565.4537587 |
| *2 | Taliban | 520.0743769 |
| 3 | Revolutionary Armed Forces of Colombia (FARC) | 301.6771192 |
| 4 | Liberation Tigers of Tamil Eelam (LTTE) | 185.6684919 |
| 5 | Algerian Islamic Extremists | 105.3723567 |
| 6 | New People's Army (NPA) | 94.40018191 |
| 7 | Chechen Rebels | 79.78810922 |
| 8 | Salafist Group for Preaching and Fighting (GSPC) | 77.6962365 |
| 9 | Maoists | 72.64253711 |
| 10 | Al-Qa`ida in Iraq | 69.09698164 |

*The police killing query results demonstrate a noticeable difference in the determination of the top ranked group. The relevancy weighted method determines the top group as the Taliban, whereas the count-based and expertise weighted methods define the top group to be the Communist Party of India Maoist (CPI-M). I will cross-reference popular sources to establish which ranking has greater support.

Due to civil war fighting in Afghanistan the Taliban have killed many government officials, including police officers [71]. Political leader Ahmad Shah Massoud trained police forces to specifically protect civilians and keep order during troubled times with the Taliban's uprising. In 2011 the Taliban conducted targeted killings against anti-Taliban leaders including two police chiefs [72].

The CPI-M is a political party in India that aims to overthrow the government through a people's war, where the majority population draws the enemy into the interior and bleeds them dry through guerrilla warfare [73]. The CPI-M are charged as running an extortion in guise of a

popular revolution by securing lands rich in minerals, and extorting money from private companies to operate there [74]. They have been accused of keeping their area under control by violent force; blowing up government property such as railways and schools [74]. The CPI-M intimidate their political enemies by abducting and killing government officials and police officers [74]. In 2012 the Indian government deployed 100,000 paramilitary, taken from various police forces, to combat the CPI-M [75].

Popular sources indicate both groups participate in targeted police killings, however the CPI-M are clearly more focused in this regard with the aim of overthrowing the government through lethal force.

### *4.4.5 "Government Assassination" Query*

**Table 27 - count-based ranking for query "government assassination"**

| Rank | Terrorist Group | Count |
|---|---|---|
| 1 | Taliban | 795 |
| *2 | Liberation Tigers of Tamil Eelam (LTTE) | 311 |
| 3 | Communist Party of India - Maoist (CPI-M) | 167 |
| 4 | Algerian Islamic Extremists | 162 |
| 5 | Al-Shabaab | 136 |
| 6 | New People's Army (NPA) | 119 |
| 7 | Revolutionary Armed Forces of Colombia (FARC) | 105 |
| 8 | Moro Islamic Liberation Front (MILF) | 102 |
| 9 | Chechen Rebels | 96 |
| 10 | Salafist Group for Preaching and Fighting (GSPC) | 95 |

**Table 28 – relevancy weighted ranking for query “government assassination"**

| Rank | Terrorist Group | Score |
|---|---|---|
| 1 | Taliban | 198.8387922 |
| *2 | Liberation Tigers of Tamil Eelam (LTTE) | 69.76081175 |
| 3 | Algerian Islamic Extremists | 41.572419 |
| 4 | Communist Party of India - Maoist (CPI-M) | 35.28050004 |
| 5 | Al-Shabaab | 35.01804692 |
| 6 | New People's Army (NPA) | 26.57144719 |
| 7 | Salafist Group for Preaching and Fighting (GSPC) | 25.8593641 |
| 8 | Revolutionary Armed Forces of Colombia (FARC) | 23.41460417 |
| 9 | Al-Qa`ida in the Arabian Peninsula (AQAP) | 23.23828234 |
| 10 | Moro Islamic Liberation Front (MILF) | 23.18917433 |

**Table 29 - expertise weighted ranking for query “government assassination"**

| Rank | Terrorist Group | Score |
|---|---|---|
| 1 | Taliban | 591.9731932 |
| *2 | Liberation Tigers of Tamil Eelam (LTTE) | 233.144644 |
| 3 | Communist Party of India - Maoist (CPI-M) | 130.8232584 |
| 4 | Algerian Islamic Extremists | 123.9063582 |
| 5 | Al-Shabaab | 95.53870189 |
| 6 | New People's Army (NPA) | 88.29163646 |
| 7 | Revolutionary Armed Forces of Colombia (FARC) | 82.67065115 |
| 8 | Salafist Group for Preaching and Fighting (GSPC) | 75.67486157 |
| 9 | Tehrik-i-Taliban Pakistan (TTP) | 72.19746761 |
| 10 | Moro Islamic Liberation Front (MILF) | 71.83623731 |

The government assassination results contain many of the same groups as the police killing results, which isn't a surprise due to the similarity in context between the queries. *However one significant variance between the two result sets is the high expertise score of the Liberation Tigers of Tamil Eelam (LTTE), placing them in second place across all three methods.

Cross-referencing popular sources supports LTTE's expertise in assassinations. A the height of its power the LTTE executed many high-profile government officials including two world leaders; Sri Lankan President Ranasinghe Premadasa in 1993 and former Indian Prime Minister Rajiv Gandhi in 1991 [76]. Following is a table illustrating the number of assassinations and ranking of government officials assassinated by the LTTE,

**Table 30- political figures assassinated by LTTE [77]**

| Position/Status | Number |
|---|---|
| President of Sri Lanka | 1 |
| Ex Prime Minister of India | 1 |
| Presidential candidate | 1 |
| Leaders of political parties | 10 |
| Cabinet Ministers | 7 |
| Members of Parliament | 37 |
| Members of Provincial Councils | 6 |
| Members of Pradeshiya Sabha | 22 |
| Political Party Organisers | 17 |
| Mayors | 4 |

The result ranking supports the system’s ability to differentiate skills and targets and rank groups according to expertise.

### *4.4.6 "Sniper Attack" Query*

**Table 31 - count-based ranking for query "sniper attack"**

| Rank | Terrorist Group | Count |
|---|---|---|
| *1 | Black Nationalists | 7 |
| 2 | Revolutionary Armed Forces of Colombia (FARC) | 3 |
| 3 | Al-Qa`ida in the Arabian Peninsula (AQAP) | 2 |
| 4 | U/I Snipers | 1 |
| 4 | Future movement (Lebanon) | 1 |
| 4 | Palestinians | 1 |
| 4 | Liberation Tigers of Tamil Eelam (LTTE) | 1 |
| 4 | Liberation Army for Presevo, Medvedja and Bujanovac | 1 |
| 4 | Popular Resistance Committees | 1 |
| 4 | Army of God | 1 |
| 4 | Mahdi Army | 1 |
| 4 | Haqqani Network | 1 |
| 4 | Hamas (Islamic Resistance Movement) | 1 |
| 4 | Kosovo Liberation Army (KLA) | 1 |
| 4 | Republic of New Afrika | 1 |

**Table 32 - relevancy weighted ranking for query "sniper attack"**

| Rank | Terrorist Group | Score |
|---|---|---|
| *1 | Black Nationalists | 6.0253723 |
| 2 | Revolutionary Armed Forces of Colombia (FARC) | 2.42358047 |
| 3 | Al-Qa`ida in the Arabian Peninsula (AQAP) | 1.88234465 |
| 4 | U/I Snipers | 1.1709052 |
| 5 | Future movement (Lebanon) | 1.1227446 |
| 6 | Palestinians | 1.0724738 |
| 7 | Liberation Tigers of Tamil Eelam (LTTE) | 0.9804899 |
| 8 | Liberation Army for Presevo, Medvedja and Bujanovac | 0.9356205 |
| 8 | Popular Resistance Committees | 0.9356205 |
| 9 | Army of God | 0.86133003 |
| 9 | Mahdi Army | 0.86133003 |
| 10 | Haqqani Network | 0.7454073 |
| 11 | Hamas (Islamic Resistance Movement) | 0.67764115 |
| 11 | Kosovo Liberation Army (KLA) | 0.67764115 |
| 12 | Republic of New Afrika | 0.57499754 |

**Table 33 - expertise weighted ranking for query “sniper attack"**

| Rank | Terrorist Group | Score |
|---|---|---|
| *1 | Black Nationalists | 7.665488567 |
| 2 | Revolutionary Armed Forces of Colombia (FARC) | 2.350118424 |
| 3 | Republic of New Afrika | 1.960262891 |
| 4 | Al-Qa`ida in the Arabian Peninsula (AQAP) | 1.566745616 |
| 5 | U/I Snipers | 0.783372808 |
| 5 | Future movement (Lebanon) | 0.783372808 |
| 5 | Palestinians | 0.783372808 |
| 5 | Liberation Tigers of Tamil Eelam (LTTE) | 0.783372808 |
| 5 | Liberation Army for Presevo, Medvedja and Bujanovac | 0.783372808 |
| 5 | Popular Resistance Committees | 0.783372808 |
| 5 | Mahdi Army | 0.783372808 |
| 5 | Haqqani Network | 0.783372808 |
| 5 | Hamas (Islamic Resistance Movement) | 0.783372808 |
| 5 | Kosovo Liberation Army (KLA) | 0.783372808 |
| 6 | Army of God | 0.2 |

The query sniper attack did not generate nearly as many results as the previous queries. I believe the excerpts from the attacks indexed do not usually discern the use of sniper rifles, and therefore result in much fewer hits. The low level of scoring also indicates that the count-based method can result in many ties.

*The ranking places Black Nationalists as number one across all three methods. Black Nationalists represent individuals and groups that urge separatism from European society, and support the establishment of a self-governing Black community [78]. The majority of Black Nationalist members reside in the United States, and would have access to high tech weapons like modern sniper rifles. Successful attacks occurring in the United States are typically discrete, due to the high level of security, and consequently excerpts from these attacks would detail the use of snipers, whereas a more brazen attack in another country may dismiss details like this.

# Chapter Five: **Conclusion**

This work demonstrates the effectiveness of applying enterprise search technology, best practices, and evaluation measures towards expertise location. Results indicate expertise can be located with the correct usage of information extraction and retrieval techniques, a good repository of information, and an expertise calculation method that takes into account quality and frequency of contextual evidence.

Development of the first terrorist expertise locator TEL showcases the importance system configuration and tuning play during performance measure testing to ensure relevant search results. The process of tuning and configuration needs to be driven by the structure of the content, which can be realized through data analysis. Achieving the highest possible relevancy in the search engine is critical to obtaining good expertise rankings.

Three expertise calculations were compared, in a series of query-based tests, to determine which produced terrorist group rankings that align closest to the sentiment of domain experts from popular media and government agencies; Times, The Daily Telegraph, The Weekly, The New York Times, and agency reports from the FBI and US Department of Defense. After analyzing the rankings and investigating their deviations by cross-referencing, it is certain that the expertise weighted method generates the best ranking. The count-based method is simple and effective, but it frequently generates scenarios where groups have a tied score. The relevancy weighted method also performs well in determining expertise, but its ignorance of attack quality produces inferior rankings to the expertise weighted method.

## 5.1 Future Work

The framework, methods, and practices used in building TEL can be re-purposed for other domains requiring expertise location. One potential use I have recognized is to assist with the identification of medical specialists for patients. I foresee general practitioners as the users for such a system, giving them the ability to quickly find the right specialist to address their patient's health issue. I initially wanted to develop this solution in place of TEL, however access to patient records and consulting reports needed to drive such a system were not available.

Visualization of results from expertise location systems is another area I would like to explore. I am interested in finding unique and effective ways of conveying snippets of the relevant evidence that is used in calculating an expertise score. A graph based visual may lend itself nicely to result presentation, especially if relationships are present in the dataset.

## References


[1] Lewis, Bob. "Guest Editor's Introduction: A Glimpse at the Future of Enterprise Search." *IT professional* (2007): 12-13.

[2] Bean, David. "How Advances in Search Combine Databases, Sentence Diagramming, and." *IT professional* 9.1 (2007): 14-19.

[3] Knabe, Frederick, and Daniel Tunkelang. "Enterprise information access and the user experience." *IT professional* 9.1 (2007): 21-28.

[4] Buxton, Stephen. "Beyond search: content applications." *IT professional* 9.1 (2007): 29-35.

[5] Yuhanna, Noel. "Today's challenge in government: What to do with unstructured information and why doing nothing isn't an option." *Forrester Research. Nov* (2010).

[6] "Enron Email Dataset". *Cs.cmu.edu*. Web. 12 Oct. 2011. http://www.cs.cmu.edu/~enron/

[7] Hawking, David. "Challenges in enterprise search." *Proceedings of the 15th Australasian database conference-Volume 27*. Australian Computer Society, Inc., (2004): 15-24

[8] Feldman, Susan, and Chris Sherman. *The high cost of not finding information*. Information Today, Incorporated, 2004.

[9] Bailey, Peter, David Hawking, and Brett Matson. "Secure search in enterprise webs: tradeoffs in efficient implementation for document level security." *Proceedings of the 15th ACM international conference on Information and knowledge management*. ACM, (2006): 493-502

[10] Keim, Brandon. "News feature: WikiMedia." *Nature Medicine* 13.3 (2007): 231-233.

[11] "Apache Lucene – Apache Solr". *lucene.apache.org*. Web. 1 Sep. 2011. http://lucene.apache.org/solr/

[12] Lenzerini, Maurizio. "Data integration: A theoretical perspective." *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2002.

[13] Dubie, Denise. "Time spent searching cuts into company productivity." *Network World.*Web. 2 Sep. 2011. http://www.networkworld.com/news/2006/102006-search-cuts-productivity.html

[14] Halevy, Alon Y., et al. "Enterprise information integration: successes, challenges and controversies." *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. ACM, 2005.

[15] Gaizauskas, Robert, and Yorick Wilks. "Information extraction: Beyond document retrieval." *Journal of documentation* 54.1 (1998): 70-105.

[16] Brin, Sergey, and Lawrence Page. "The anatomy of a large-scale hypertextual Web search engine." *Computer networks and ISDN systems* 30.1 (1998): 107-117.

[17] Xu, Xiaomei, et al. "Improving document search by finding domain experts."*Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on*. Vol. 3. IEEE, (2009): 270-273

[18] Chen, Wu, Wei Lu, and Shuguang Han. "Designing and implementation of expertise search & hotspot detecting system." *Networked Computing (INC), 2010 6th International Conference on*. IEEE, (2010): 1-5

[19] “Google Enterprise Search”. *Google.com*. Web. 14 Oct. 2011. ”http://www.google.com/enterprise/search/index.html

[20] Perrin, Chad. "The CIA Triad". *Techrepublic.com.* Web. 24 Oct. 2011. http://www.techrepublic.com/blog/security/the-cia-triad/488

[21] “Google Search Appliance Security”. *static.googleusercontent.com.* Web. 13 Nov. 2011. http://static.googleusercontent.com/external_content/untrusted_dlcp/www.google.com/en//support/enterprise/static/gsa/docs/admin/en/GSASecurity.pdf

[22] Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Vol. 1. Cambridge: Cambridge University Press, 2008.

[23] Guha, Ramanathan, Rob McCool, and Eric Miller. "Semantic search."*Proceedings of the 12th international conference on World Wide Web*. ACM, (2003): 700-709

[24] Morris, Meredith Ringel. "Collaborative search revisited." *Proceedings of the 2013 conference on Computer supported cooperative work*. ACM, 2013.

[25] Grimes, Seth. "Breakthrough analysis: two+ nine types of semantic search." *InformationWeek* (2010).

[26] Macdonald, Craig, David Hannah, and Iadh Ounis. "High quality expertise evidence for expert search." *Advances in Information Retrieval*. Springer Berlin Heidelberg, (2008): 283-295.

[27] Macdonald, Craig, and Iadh Ounis. "Voting for candidates: adapting data fusion techniques for an expert search task." *Proceedings of the 15th ACM international conference on Information and knowledge management*. ACM, 2006.

[28] Amati, Giambattista. "Frequentist and bayesian approach to information retrieval." *Advances in Information Retrieval*. Springer Berlin Heidelberg, 2006. 13-24.

[29] Macdonald, Craig, and Iadh Ounis. "Voting techniques for expert search."*Knowledge and information systems* 16.3 (2008): 259-280.

[30] Macdonald, Craig, and Iadh Ounis. "Searching for expertise: Experiments with the voting model." *The Computer Journal* 52.7 (2009): 729-748.

[31] Balog, Krisztian, Leif Azzopardi, and Maarten De Rijke. "Formal models for expert finding in enterprise corpora." *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2006.

[32] Davenport, Thomas H., and Laurence Pruzak. *Working knowledge: How organizations manage what they know*. Harvard Business Press, 2000.

[33] "Text Retrieval Conference". *Trec.nist.gov.* Web. 11 Dec. 2011. http://trec.nist.gov/

[34] "Performance and Correctness Measures". *en.wikipedia.org.* Web. 7 Jan. 2012. http://en.wikipedia.org/wiki/Information_retrieval#Performance_and_correctness_measures

[35] Järvelin, Kalervo, and Jaana Kekäläinen. "IR evaluation methods for retrieving highly relevant documents." *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, (2000): 41-48.

[36] Zhu, Mu. "Recall, precision and average precision." *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo* (2004).

[37] "Precision and Recall". *en.wikipedia.org.* Web. 9 Jan. 2012. http://en.wikipedia.org/wiki/Precision_and_recall

[38] "Evaluation of Ranked Retrieval Results". *nlp.stanford.edu.* Web. 11 Jan. 2012. http://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-ranked-retrieval-results-1.html

[39] National Consortium for the Study of Terrorism and Responses to Terrorism (START). (2012). Global Terrorism Database [Data file]. Retrieved from http://www.start.umd.edu/gtd

[40] Erin Miller; Kathleen Smarick, 2013, "Profiles of Perpetrators of Terrorism in the United States (PPT-US)", http://hdl.handle.net/1902.1/17702 UNF:5:VrK7t3x1k/CQjRHpkuPzdg== National Consortium for the Study of Terrorism and Responses to Terrorism [Distributor] V5 [Version]

[41] Auburn, David. *Oxford American writer's thesaurus*. Oxford University Press, 2004.

[42] “The Porter Stemming Algorithm”. *tartarus.org.* Web. 11 May. 2012.

http://tartarus.org/martin/PorterStemmer/

[43] “Reddit – The Front Page of the Internet”. *Reddit.com*. Web. 21 Sep. 2012. http://reddit.com

[44] "What is meaning-based computing". *www.computing.co.uk.* Web. 8 Oct. 2012.

http://www.computing.co.uk/itweek/analysis/2162661/interview-meaning-computing

[45] “HP to acquire Autonomy HP Newsroom”. *hp.com*. Web. 18 Nov. 2012.

http://www.hp.com/hpinfo/newsroom/press/2011/110818xc.html

[46] “Google – Our History in Depth”. *Google.ca.* Web. 6 Dec. 2012.

http://www.google.ca/about/company/history/

[47] “Google Appliance Now Searching by the Billion”. *Cnet.com*. Web. 7 Dec. 2012.

http://news.cnet.com/8301-1001_3-10254486-92.html

[48] “Google Releases New Version of its Search Appliance”. *Computerworld.com.* Web. 22 Jan. 2013.

http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9133839

[49] “Solr CNET Code Contribution”. *Apache.org.* Web. 3 Feb. 2013.

http://issues.apache.org/jira/browse/SOLR-1

[50] “LinkedIn”. *Linkedin.com.* Web. 6 Feb. 2013. http://www.linkedin.com

[51] "Taliban and the Northern Alliance". *US Gov Info. About.com*. Web. 11 Feb. 2013.

http://usgovinfo.about.com/library/weekly/aa092801a.htm

[52] “Columbian Soldiers Die in Clashes”. *Bbc.co.uk.* Web. 22 Feb. 2013.

http://www.bbc.co.uk/news/world-latin-america-23394408

[53] "Sri Lankan experience proves nothing is impossible". *sundayobserver.lk*. Web. 5 Mar. 2013. http://www.sundayobserver.lk/2011/06/05/sec03.asp

[54] Ben Arnoldy. "In Afghanistan, Taliban kills more civilians than US". *Csmonitor.com.* Web. 16 Mar. 2013. http://www.csmonitor.com/2009/0731/p06s15-wosc.html

[55] “General Sir Michael Jackson: We must maintain our will in Afghanistan". *Telegraph.co.uk.* Web. 21 Mar. 2013. http://www.telegraph.co.uk/news/newstopics/onthefrontline/2171923/General-Sir-Michael-Jackson-We-must-maintain-our-will-in-Afghanistan.html

[56] "Citing rising death toll, UN urges better protection of Afghan civilians". *Web.archive.org*. Web. 27 Mar. 2013. http://web.archive.org/web/20110726085402/http://unama.unmissions.org/Default.aspx?tabid=1783&ctl=Details&mid=1882&ItemID=12602

[57] Haddon, Katherine. "Afghanistan marks 10 years since war started". *Web.archive.org*. Web. 2 Apr. 2013. http://web.archive.org/web/20111010055026/http://news.yahoo.com/afghanistan-marks-10-years-since-war-started-211711851.html

[58] "UN: Taliban Responsible for 76% of Deaths in Afghanistan". *Weeklystandard.com*. Web. 5 Apr. 2013. http://www.weeklystandard.com/blogs/taliban-responsible-76-deaths-afghanistan-un

[59] "February 1993 Bombing of the World Trade Center in New York City". *Web.archive.com.* Web. 14 Apr. 2013. http://web.archive.org/web/20061207130217/http://cns.miis.edu/pubs/reports/wtc93.htm

[60] "Bin Laden claims responsibility for 9/11". *Cbc.ca.* Web. 26 Apr. 2013. http://www.cbc.ca/world/story/2004/10/29/binladen_message041029.html

[61] Gopal, Anand. "The most deadly US foe in Afghanistan". *Csmonitor.com*. Web. 5 May 2013. http://www.csmonitor.com/World/Asia-South-Central/2009/0601/p10s01-wosc.html

[62] Robert A. Pape; Lindsey O'Rourke and Jenna McDermit."What Makes Chechen Women So Dangerous?". *The New York Times*. Web. 6 May 2013. http://www.nytimes.com/2010/03/31/opinion/31pape.html?pagewanted=1&emc=eta1

[63] "Seven dead in Chechnya suicide bombing". *Abc.net*. Web. 15 May 2013. http://www.abc.net.au/news/stories/2009/07/27/2636848.htm

[64] "Death Toll Rises in Ingush Suicide Blast". *The Moscow Times*. Web. 15 May 2013. http://www.moscowtimes.ru/article/1010/42/380929.htm

[65] "Guerrilla miners". *The Economist*. Web. 4 Jun. 2013 http://www.economist.com/node/18013780

[66] Livingstone, Grace. *Inside Colombia: drugs, democracy and war*. Rutgers University Press, 2003.

[67] Rommel C. Banlaoi. "Abu Sayyaf Group: From Mere Banditry to Genuine Terrorism". *Muse.jhu.edu.* Web. 9 Jun. 2013. http://muse.jhu.edu/journals/southeast_asian_affairs/summary/v2006/2006.banlaoi.html

[68] "Funding Terrorism in Southeast Asia: The Financial Network of Al Qaeda and Jemaah Islamiyah". *The National Bureau of Asian Research*. Web. 18 Jun. 2013. http://www.nbr.org/publications/analysis/pdf/vol14no5.pdf

[69] "Abu Sayyaf kidnappings, bombings and other attacks". *GMA News*. Web. 24 Jun. 2013. http://www.gmanews.tv/story/154797/abu-sayyaf-kidnappings-bombings-and-other-attacks

[70] Chouvy, Pierre-Arnaud. *Opium: Uncovering the politics of the poppy*. Harvard University Press, 2009.

[71] "Blood-Stained Hands, Past Atrocities in Kabul and Afghanistan's Legacy of Impunity". *Human Rights Watch*. Web. 2 Jul. 2013. http://www.hrw.org/en/reports/2005/07/06/blood-stained-hands

[72] “OARDEC (2008). Unclassified Summary of Administrative Review Board Proceedings”. *United States Department of Defense*, pp. 685–90. Web. 3 Jul. 2013. http://dspace.wrlc.org/doc/get/2041/66914/03096display.pdf

[73] "Communist Party of India-Maoist (CPI-Maoist)". *South Asia Terrorism Portal*. Institute for Conflict Management. Web. 7 Jul. 2013. http://www.satp.org/satporgtp/countries/india/terroristoutfits/CPI_M.htm

[74] Robinson, Simon. "India's Secret War". *Time*. Web. 19 Jul. 2013. http://www.time.com/time/magazine/article/0,9171,1810169-1,00.html

[75] Sharma, Aman. "New crack Greyhound commando forces to be deployed in five more Maoist-affected states". *Daily Mail*. Web. 21 Jul. 2013. http://www.dailymail.co.uk/indiahome/indianews/article-2174049/Crack-Greyhound-commando-forces-deployed-Maoist-affected-states.html

[76] "Taming the Tamil Tigers". *Federal Bureau of Investigation*. Federal Bureau of Investigation. Web. 26 Jul. 2013. http://www.fbi.gov/page2/jan08/tamil_tigers011008.html

[77] "Humanitarian Operation – Factual Analysis, July 2006 – May 2009". *Ministry of Defence (Sri Lanka)*. Web. 26 Jul. 2013. http://www.defence.lk/news/20110801_Conf.pdf

[78] “Black Nationalism”. *Stanford.edu.* Web. 3 Aug. 2013. http://mlk-kpp01.stanford.edu/index.php/encyclopedia/encyclopedia/enc_black_nationalism

[79] “Autonomy”. *Autonomy.com.* Web. 9 Aug. 2013. http://www.autonomy.com

[80] “Unlocking the Value of Your Information”. *Intergen.co.nz.* Web. 10 Aug. 2013. http://www.slideshare.net/IntergenNZ/unlocking-the-value-of-your-information

[81] “OpenPipeline”. *Openpipeline.com.* Web. 11 Aug. 2013. http://openpipeline.com/

[82] “Solr Wiki - Analyzers, Tokenizers, and Token Filters”. *Wiki.apache.org.* Web. 18 Aug. 2013. http://wiki.apache.org/solr/AnalyzersTokenizersTokenFilters

[83] “Solr Wiki – Relevancy FAQ”. *Wiki.apache.org.* Web. 18 Aug. 2013. http://wiki.apache.org/solr/SolrRelevancyFAQ

[84] “Transport Layer Security”. *Wikipedia.org.* Web. 1 Sep. 2013. http://en.wikipedia.org/wiki/Secure_Sockets_Layer

[85] “Zappos”. *Zappos.com.* Web. 6 Sep. 2013. http://www.zappos.com/

[86] “Amazon”. *Amazon.com.* Web. 7 Sep. 2013. http://www.amazon.com/

[87] “Comment Tag Clouds”. *W-shadow.com.* Web. 11 Sep. 2013. http://w-shadow.com/blog/2008/11/05/digg-vs-reddit-comment-tag-clouds/