UNF Digital Commons

UNF Graduate Theses and Dissertations

Student Scholarship

2017

Improving Search Ranking Using a Composite Scoring Approach

Larry D. Snedden

Improving Search Ranking Using a Composite Scoring Approach

by

Larry D. Snedden

A thesis submitted to the
School of Computing
in partial fulfillment of the requirements for the degree of

Master of Science in Computer and Information Sciences

UNIVERSITY OF NORTH FLORIDA
SCHOOL OF COMPUTING

December, 2017

The thesis “Improving Search Ranking Using a Composite Scoring Approach” submitted by Larry D. Snedden in partial fulfillment of the requirements for the degree of Master of Science in Computer and Information Sciences has been

Approved by the thesis committee: Date

________________________________ ________________

Dr. Sherif A. Elfayoumy
Thesis Advisor and Committee Chairperson

________________________________ ________________

Dr. Robert F. Roggio
Thesis Committee Member

________________________________ ________________

Dr. Karthikeyan Umapathy
Thesis Committee Member

Accepted for the School of Computing:

________________________________ ________________

Dr. Sherif A. Elfayoumy
Director of the School

Accepted by the College of Computing, Engineering, and Construction:

________________________________ ________________

Dr. Mark Tumeo
Dean of the College

Accepted by the University:

________________________________ ________________

Dr. John Kantner
Dean of the Graduate School

ACKNOWLEDGEMENT

I wish to thank my family, friends, and colleagues for their continuous support in achieving this accomplishment in my education and career as an Information Sciences Professional. I would also like to extend my gratitude to the Fusion Center of Jacksonville for allowing me to work with them in the development of this thesis.

CONTENTS

FIGURES

TABLES

EQUATIONS

ABSTRACT

In this thesis, the improvement to relevance in computerized search results is studied. Information search tools return ranked lists of documents ordered by the relevance of the documents to the user supplied search. Using a small number of words and phrases to represent complex ideas and concepts causes user search queries to be information sparse. This sparsity challenges search tools to locate relevant documents for users. A review of the challenges to information searches helps to identify the problems and offer suggestions in improving current information search tools. Using the suggestions put forth by the Strategic Workshop on Information Retrieval in Lorne (SWIRL), a composite scoring approach (Composite Scorer) is developed. The Composite Scorer considers various aspects of information needs to improve the ranked results of search by returning records relevant to the user's information need.

The Florida Fusion Center (FFC), a local law enforcement agency has a need for a more effective information search tool. Daily, the agency processes large amounts of police reports typically written as text documents. Current information search methods require inordinate amounts of time and skill to identify relevant police reports from their large collection of police reports.

An experiment conducted by FFC investigators contrasted the composite scoring approach against a common search scoring approach (TF/IDF). In the experiment, police

investigators used a custom-built software interface to conduct several use case scenarios for searching for related documents to various criminal investigations. Those expert users then evaluated the results of the top ten ranked documents returned from both search scorers to measure the relevance to the user of the returned documents. The evaluations were collected and measurements used to evaluate the performance of the two scorers. A search with many irrelevant documents has a cost to the users in both time and potentially in unsolved crimes. A cost function contrasted the difference in cost between the two scoring methods for the use cases. Mean Average Precision (MAP) is a common method used to evaluate the performance of ranked list search results. MAP was computed for both scoring methods to provide a numeric value representing the accuracy of each scorer at returning relevant documents in the top-ten documents of a ranked list of search results.

The purpose of this study is to determine if a composite scoring approach to ranked lists, that considers multiple aspects of a user’s search, can improve the quality of search, returning greater numbers of relevant documents during an information search. This research contributes to the understanding of composite scoring methods to improve search results. Understanding the value of composite scoring methods allows researchers to evaluate, explore and possibly extend the approach, incorporating other information aspects such as word and document meaning.

Chapter 1

INTRODUCTION

“Information Retrieval (IR) is the science of searching for information in documents of an unstructured nature” (Han12). One aspect of IR is a search, which is often the interface between the information seeker and the discipline of IR. The ubiquity of the Internet and its many web search engines provides many familiar examples where a search is conducted and relevant documents are returned to the user in a ranked order, called ranked lists. These lists are organized, or computed by various scoring methods (NLP14). Because the tools of the Internet are specialized for use in a public domain and because of intellectual property concerns, legal and sometimes secretive requirements, businesses, doctors, law enforcement and others are unable to publish sensitive information on the Web to make effective use of web search tools. This has provided both the commercial and open source community motivation to develop IR search tools enabling various businesses, agencies, and users to create solutions designed especially for their information needs (Allan12). Several resources are available to researchers and developers; Searching On Lucene with Replication (SOLR), Lucene, and WordNet are just a few popular examples (Whissel09). The availability of these tools is making the development of effective expert search tools possible for traditionally underserved domains. These disparate domains have needs for search systems to search large collections of documents and data enabling them to produce invaluable information.

Document similarity is the branch of Information Retrieval (IR) that measures the relevance between delineable units of information (Grefenstette09). The most common methods of measuring these similarities are based on vector space models whereby a mathematical operation based on the number of terms in a document are represented as vectors. The angularity between the vectors are computed and used to measure how similar or close they are to one another (Sanderson12). This is referred to as cosine similarity. Other approaches to using the vector space model consider the magnitude of the computed vectors and not just the angles. A set of documents (corpus) can be represented as a set of vectors where there exists an axis for each term, rather than an axis for the collection of terms. Each term can now play a significant role in computing the overall measure of the document. Terms, which are infrequent in the corpus of documents, will have more significance in computing the vector space score for a document than those, which occur frequently (NLP14). A very common scoring method is Term Frequency Inverted Document Frequency (TF/IDF). IR search systems, provide ranked lists of documents in response to user search queries. These queries are terms, or phrases consisting of “keywords” (Allan12). TF/IDF computes a vector score used for comparison only instead of comparing documents; it compares a search query to documents in the corpus, scoring documents for similarity to the query terms. TF/IDF is described in detail in Chapter 2.

A common challenge for IR is that of sparsity (Demers15). Sparsity in this context is the sparsity of information contained in user search queries, which are made up of a few terms or phrases compared to the documents in a corpus. The small amount of terms supplied by a person to describe a complex information need make it challenging for an IR System to locate the relevant information. Where a large document is rich with information, queries are generally short by comparison and so when using vector space comparisons between a query and documents contained in a corpus, there is a challenge to capture the user’s intent or concept. When a user creates an information query, they have any number of concepts in mind and select terms to express them. Query expansion, where information is added to a sparse query is accomplished by adding synonyms with the aid of a thesaurus, or to add emphasis to chosen terms using boost or weights (Manning08). Boost and Weighting methods are numerical values applied in scoring algorithms to add emphasis to selected terms, usually acting as a multiplier. Both of these methods can affect precision and recall, two performance metrics of IR systems. Precision measures the “exactness”, or the percentage of documents identified as positive that are indeed positive. Recall is the measure of “completeness”, what percentage of positive items are identified as positive (Han12). Precision (See Figure 1) is the area where retrieved documents are relevant, making them True Positives (TP). Recall (See Figure 1) includes documents incorrectly returned as relevant when they are in fact not. These are False Positives (FP).

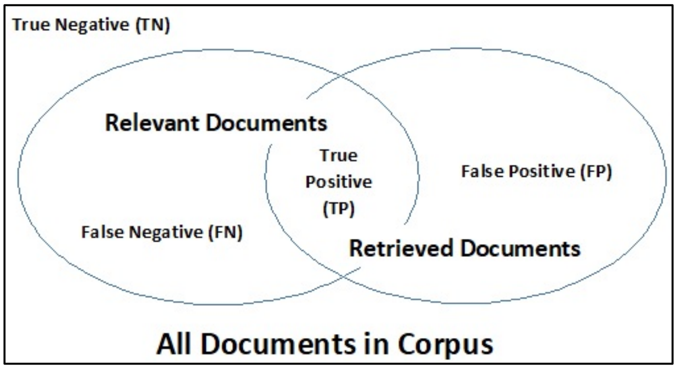


Figure 1: Venn Diagram of Precision and Recall.

The use of thesauri or ontologies generally increases the recall of search, where more documents are found because of the query expansion. High recall rates do not necessarily mean an increase in accuracy or precision of the document search results for which the user is searching (Manning08). Precision would measure how many items are relative in a collection of returned items while Recall would be how complete the collection of returned items is. Boosting a search does not cause an increase in recall but is a form of query expansion (Manning08). Therefore, boosting may have a larger improvement on precision and less of an effect on recall thus making it an item of focus for improvement in ranked retrieval IR systems. Accuracy measurements are different than precision and recall measurements because accuracy will not give us specifics on how well a classifier identifies true positive (Sensitivity) and false positives (Specificity). Precision and Recall allows for the calculations of Specificity and Sensitivity (Han12).

1.1 Problem Description

Much of IR systems development and research has focused on improving the precision and recall of search tools. An important problem of the IR process is that users and their expertise are often neglected (Belkin08). User expertise in a domain is extremely important to successfully performing a document search using IR systems. The Strategic Workshop on Information Retrieval in Lorne (SWIRL) identified this importance and published the following objective needs enumerated in Table 1 (Allan12).

Requirements for effective IR Search	
A	Not Just Ranked List: Consider Enriched Query Methods.
B	Help For Users: Develop Methods, Which Make IR Search Easier For Untrained Users.
C	Capturing Context: Incorporate User’s Individual Context.
D	Domains: Consider Information Needs In Restricted Domains.
E	Using Structure: Integration of Document Structure.

Table 1: Search Requirements

IR search systems are built expressly to assist users in making accurate and timely analyses. One of the problems found in the literature that remains unsolved for these systems is they fail to take in the expertise of the user in identifying important information for the retrieval process. By failing to correctly capture the essence of information an expert user is trying to convey in words, IR systems must resort to probabilities that the meaning in document collections matches the user need based on measures against the collection or the behaviors of other users accessing the collection (Belkin08). The Google Web Search Engine uses such an approach with the PageRank and Hyperlink Induced Topic Search (HITS) algorithms. An example of a unique concept search would be a user wanting to retrieve relevant documents containing a mix of specific words or phrases as well as semantically similar topics: “The rifle used in the robbery was a Ruger Mini-14”. If the system treats all terms of our example query equally, the result set will contain many false-positives. This is because there is a lack of context given to the search terms. Documents with the terms rifle and robbery may be widespread in the corpus while the expert user is interested in documents related to robberies that contain exacting details of a Ruger Mini-14 (a type of rifle). By more accurately capturing what the user is trying to find, a search tool can increase the accuracy of matching the needs in the ranked results. This makes it much easier for the user to discern what documents can be ignored, similar to how Internet searchers generally only view the top few webpages returned from a search even though they may have hundreds of returned documents (Manning08).

1.2 Problem Statement

Based on the evidence presented in the literature and the findings in Table 1, capturing user perspective and leveraging documents written structure in computing ranked scores should improve the accuracy and effectiveness of IR search tools over current methods. In this research, a Composite Scoring Method, which provides an increase in precision over TF/IDF vector scores, will be developed. Creating a composite score, which includes consideration of the structure of user concepts and ideas, the frequency, and the number of ideas found in a document, should provide better ranked retrieval results. Improved ranked search results reduces the time analysts spend reviewing less or non-relevant documents. This is especially important in domains where the review of every relevant document is necessary, such as law enforcement and intelligence.

Chapter 2

LITERATURE REVIEW AND BACKGROUND

“Information Retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).”(Manning08). These data are often retrieved by using a search tool. The results of searches are presented in a ranked order of relevance (*Ranked List*) for the user. These rankings utilize various scoring methods to score the documents or data in a corpus to match a search query (Allan12 and Sanderson12). These searches are usually performed upon collections of unstructured data (e.g. web pages, police reports, financial documents). As data collections increase in size, traditional cataloguing techniques become inadequate necessitating the use and development of more efficient IR systems (Sanderson12). An example of an IR search would be a simple text search using the Uniplexed Information System (UNIX) Grep tool to find a word in a single text document or file structure. The processing power of the modern computer makes this word search using a UNIX utility seem trivial, but using this simple tool on an extremely large collection of documents would make it readily apparent that this is not an efficient tool. The results of such a simple search would result in a potentially large set of only loosely related items, which would require inordinate amounts of manual inspection to find relevant or valid items from the returned collection. The exponential growth of digital information and high-speed networking has produced a common- need for better search tools. As a result, information search has become ubiquitous in our modern Information Age (Sanderson12).

2.1 Previous Work

Jhon Whissel's thesis, *Information Retrieval Using Lucene and WordNet*, presents evidence of the value of combining open source tools to provide advancements for Information Retrieval and the value of open source development (Whissel09). His incorporation of WordNet is an effort to help identify word usage and improve the comparison of similarity between documents by examining the meaning or context of terms and not simply the specific use of a term. His work addressed the part of the challenges posed in objective A, Consider Enriched Query Methods. He uses *WordNet* to perform a form of query expansion and disambiguation. To accomplish this he uses *synsets*, which are collections of words with similar meaning or usage, such as car and auto. Whissel concluded that ad-hoc searches with unigram terms, terms composed of just one word, were dramatically more accurate with the incorporation of WordNet produced synsets (Whissel09). The value of his work for this work was to affirm that indeed, adding information to user queries was an important consideration in the improvement of IR search.

Professor Laurie Hirsch developed and tested an automatic text classification tool to improve the ability to measure similarity between documents. He applied the idea of genetic adaptation to create classifiers to identify related documents in ready-to-use search query forms that are easily understood by users. The value in this work was not the performance of the classifier, but the focus on the ease of human understanding in the construction of the search query (Hirsch10). This addressed Requirement B of Table 1: Develop Ways To Make IR Easier For Untrained Users. Hirsch made the user query easy to use with little to no training. Hirsch's experiment did not require users to have any knowledge of query construction.

The growth and variety of information in the modern age, presents the need for continued research into improving the effectiveness of ad-hoc search tools for IR systems. IR search tools focus on returning a ranked or sorted list of documents to satisfy a user's query. The main differences between Web search systems and IR based search tools is that Web tools such as Google rank documents based on their prominence on the Internet using keywords while document searching in IR search systems focus on analyzing the document content regardless of its popularity. In this regard, the Internet search tools are not concerned with an expert user's intent or perspective as a part of a document search analysis.

The growth and variety of information in the modern age, presents the need for continued research into improving the effectiveness of ad-hoc search tools for IR systems. IR search tools focus on returning a ranked or sorted list of documents to satisfy a user's query or information need. There various types of IR Search: Web, Desktop, Enterprise, and Database just to name a few. The various types are generally different in the architectures and methods with which they are executed and the information domains they are intended to work in. Information Domains may refer to how documents are constructed and stored as well as whether they are public or private.

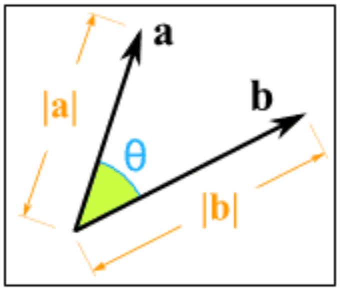
"…Google is the world's most popular search engine." (Krawczyk14). It is so common that people use the word "Google" as a verb to describe searching the Internet (Merriam14). The obvious question given the popularity of the Google Search engine is why do we need other IR tools? The simple answer to that question is the Google Search engine is specifically designed for searching for unstructured data on the Internet, a specific information domain. Additionally, the components making up the Google search engine rely on the linking of content from one website to the other, the popularity or frequency of visits to a site or page, and the ability to access the Internet twenty-four hours a day. A key part of the success of the Google Search engine is the *PageRank* algorithm. This algorithm uses a combination of linked frequency and search quantity to assign credibility to root sites on the Internet. Those sites then have a heavier weight in computing the validity or relevance of a search result (Sanderson12, Strickland14 and Brinkmeir06). For this reason, trying to implement this IR search system on a set of medical records or other offline sensitive data in the absence of these external supporting Internet links hinders the efficacy. The search tool is designed for public documents on the Internet and not for offline private text data. (Rogers17). This tool fails to meet the search requirement D of Table 1: Domains: Consider information needs in restricted domains.

2.2 Vector Space Model

For document searches, document similarity calculations are often performed using the Vector Space Model (VSM). This VSM represents text documents as vectors of terms. For example, a document *X* “The cow jumped over the moon and the fox jumped over the cow” may be represented as an array of the term frequencies of which it is comprised. The array for document *X*((the,3)(cow,2)(over,2)(moono,1)(and,1)(fox,1)) can now be compared to other documents on the Euclidean plane to mathematically compare their proximal values to one another. The closer they are to one another the more similar the information they may share.

2.2.1 Cosine Similarity

One measurement method is called Cosine Similarity. It uses the inner product space measuring the cosign angle between the two vectors. It is calculated by first creating a vector for each document, counting the term frequency for a desired set of terms in each document. These arrays, or vectors are then subjected to a Euclidean Normalization. This involves calculating the cosine similarity of the document term vectors. Equation 1 illustrates the Cosine Similarity measurement.



Sim(a, b) = |a| x |b| x cos(ө)

Equation 1: Cosine Similarity (Math17)

Documents or parts of documents referred to as collections of terms and phrases are reduced mathematically into a *term-frequency vector*. As an example, two documents and their term frequencies for three specific terms arrest, drunk, and drug are shown in Table 2.

Document	**arrest**	**drunk**	**drug**
Document 1	3	0	1
Document 2	2	1	1

Table 2: Document Term Frequencies

Document 1 can be vector x represented here as x (3, 0, 1) while Document 2 can be represented as y (2, 1, 1). These two vectors are combined computing an *inner-dot product*. $\vec{V}(x)\times\vec{V}(y) = (3\times2 + 0\times1 + 1\times1) = 7$. Now the *absolute value* of the vectors is determined: $||x|| = \sqrt{3^2 + 0^2 + 1^2}$ = 3.16, $||y|| = \sqrt{2^2 + 1^2 + 1^2}$ = 2.45. These two absolute values are multiplied and divided by the *inner-dot product* shown here: $7/(3.16 * 2.45)$ = 7 / 7.742 = .90. The closer the number is to one, the more similar the two documents are while the closer they are to zero means they are exactly orthogonal to each other with no similarity at all. This allows for a ranking of documents by a numerical score based on how similar they are to each other or to a search query represented as a term frequency vector.

One drawback of using cosine similarity in searches is that it has no scheme to apply weight or boost to rare terms in the document vectors. All terms in documents are equally weighted regardless of the number of terms in the individual documents or in the Corpus.

2.2.2 TF/IDF

Because Cosine Similarity does not consider the importance of any term over any other terms in its calculation, a new method was developed. While Cosine Similarity allowed for measuring an angular relationship between term vectors, it does not consider magnitude. Term Frequency-Inverse Document Frequency (TF/IDF) adds the ability to weight documents based upon the term frequency contained in them and increase the weight of terms appearing infrequently in the corpus. The idea is that Term Frequency is a measure of how important it is to the overall meaning or concept of the document that contains it. The more times a word appears in a document the more important it is to that document, but the more often it appears in the collection of documents, the less important it is overall (Hirsch10). The TF/IDF scoring method makes use of document-level statistics to apply a weight or value to the terms of a document. This is accomplished by calculating the Inverted Document Frequency (IDF). "The IDF is a measure of the relevance of a term. The higher the IDF is, the more relevant the term is." (Chen17). Given a collection of *N* documents, the IDF of a term *t* is computed using the formula shown below in Equation 2.

$$idf_t = \log\left(\frac{N}{df_t}\right)$$

Equation 2: IDF Formula (NLP14)

The IDF value is then applied to the Term Frequency (TF) vector for a document as shown below in Equation 3.

$$tf - idf_{t,d} = tf_{t,d} \times idf_t$$

Equation 3: TF/IDF Formula (NLP14)

The application of IDF gives a higher weight to those terms, which occur, frequently in a small number of documents but a lower weight when they appear in many of the documents. These TF/IDF scores allow a convenient ranking method. This method is the most widely used and well documented in IR search scoring method (Manning09 and Sanderson12).

As a simple example showing how to compute TF/IDF of a document given a search for the term "arrest" consider there is a document 1000 terms in length in the corpus using this term 4 times. The Term Frequency (TF) measures (4 / 1000) = .004. Assume there are 1 million documents in the corpus in which the term "arrest" appears in 100 of these documents. The Inverse Document Frequency (IDF) would be log ($1{,}000{,}000/100$) = 4. The TF/IDF Score for this particular document in this corpus with this search term would be .004 * 4 = .016. The scores for the other 100 documents containing “arrest” would also have scores computed, thus producing a ranked list of search results. The documents scoring highly would be considered more relevant than lower scoring documents for the search.

These scoring methods, though popular do not meet the requirement A of Table 1: Not Just Ranked List: Consider Enriched Query Methods. These scoring methods also do not meet the need of requirement C of Table 1: Capturing Context: Incorporate User’s Individual Context.

2.3 Precision and Recall

Document search algorithms measure similarity comparing user searches to the text contained in the corpus. The performance of these searches is described by using precision and recall. The precision of a system relates to how accurately an IR system search finds relevant data while recall is a measure on how many documents are found with some measure of relevance to a user search. “Precision can be thought of as a measure of exactness…whereas recall is a measure of completeness…” (Han12).

$$Precision = \frac{relevant\ documents\ \cap\ returned\ documents}{returned\ documents}$$

Equation 4: Precision (Han12)

Precision in Equation 4 shows all relevant documents in the returned documents divided by the returned documents provides the precision of the search.

$$Recall = \frac{relevant\ documents\ \cap\ returned\ documents}{relevant\ documents}$$

Equation 5: Recall (Han12)

Recall in Equation 5 is computed by dividing the number of relevant documents in the returned documents by the number of relevant documents in the corpus.

As an example of computing Precision and Recall assume a search for pictures of apples from a collection of pictures of fruit is performed. The resulting search returns eight pictures, three of which are pictures of apples while the other five pictures are of assorted red fruits, which are not apples. The precision is $\left(\frac{3}{8}\right) * 100 = 37.5\ \%$. Assume we know there are actually four pictures of apples available in the collection of pictures. The recall for this search is $\left(\frac{3}{5}\right) * 100 = 60\%$.

Using precision and recall to measure the performance of an IR search tool is a common measure. When calculating the precision and recall of Ranked Lists, it is customary and popular to use calculate a single numerical measure of performance. Because there is potentially no end to a Ranked List, a fixed number *Top(k)* is selected from the top of the list to evaluate the precision of the search result. Any number of searches is conducted and the Precision and Recall is computed for the Top(k) results of each and then they are averaged to provide a *Mean Average Precision Score.*

2.4 Word Sense Disambiguation

Languages are complex and determining what a user is searching for can be challenging. Similarity can be a direct word or phrase match, a conceptual match or a semantic match. The word “love” for example, can have an exact match in similarity with the word “love” found in another document, but conceptually “love” can be similar to any word involving an emotion including the word “hate” because there is a potential ideological connection. Semantically, “love” can be used to convey a multitude of meanings ranging from the idea of an emotion to the idea of marriage. This ambiguity in word meaning is referred to as the homonymy and polysemy problem. “Homonymy describes when two senses of a given word (or derivation) are distinct.” (Stokoe08). An example would be the word “bat”. One meaning is referring to a flying animal while the other refers to a wooden tool used in a sport. “Alternatively, polysemy describes where two senses of a word are related in that they share membership of a subsuming semantic classification.” (Stokoe08). This means just because a word shares some commonality in spelling or even use that it does not share the same meaning. Without a measure of context or word usage, there is no consistent way to measure relevance and thus to score or rank documents relevance to a search (Erk08, Stanchev12 and Chaplot14).

Using effective Word Sense Disambiguation (WSD) methods can improve both the precision and recall of IR systems (Zhong12). One of the most common approaches to determining word use is by employing sentence parsing tools and dictionaries referred to as ontologies. Ontologies provide a way to establish relationships of meanings between different words depending upon how they are used in a phrase. The conceptual meaning or use of a word or phrase can be established and used in measuring similarity between documents (Trim14).

Ontological approaches to IR search pose a new set of challenges in that in order for them to be accurate; they must pertain to a restricted domain. A second challenge to using ontologies is that they are very expensive to produce requiring domain experts to annotate may well be a large corpus (Chaplot14). The nature of ad-hoc queries is that they are performed for a specific unique, one-time use. An ontological database may not be beneficial in these instances and would be difficult to produce in a short time frame. The restricted domain represented in this thesis has no ontological database available and coupled with the high dynamism of changing information this approach to WSD is not a good fit for a potential solution for this IR search problem.

2.5 Structure

The structure of the data being searched can affect the accuracy of the IR system. Structure can help provide meaning to the use or placement of words, which can establish relationships between documents. There are three types of data: Structured, Unstructured, and Semi-Structured. Structured data is organized data that has a specified form or model. An example of structured data would be the kind of data found in an entity relational database. In this example, each data item is grouped into an entity with defined attributes making up a schema. This greatly simplifies determining relevance to a search query making search results more precise (Woord14, Primmer14 and Egnor14). In stark contrast, unstructured data contains no identifying mechanism other than the word usage with which to identify any characteristics about the data. Because of the lack of information about the text, the meaning or use of these words that make up portions of the document is ambiguous. There is no rule on usage enforced and no assurance of consistency making searches more difficult (Woord14 and Trim14). Semi-structured data is that which has enough information to be grouped in some consistent way (Woord14). An example of semi-structured data would be the use of XML or other meta-language to identify elements of a document that share a common name, association or concept. Another example might be where a document contains a section “Crime Type” where an investigator might find clues about the nature of the entirety of a document based off of this small section.

There are other forms of structure in documents. The style with which a document is written contains structure as well. Scott Francis describes six writing styles: Categorical, Evaluative, Chronological, Comparative, Sequential, and Causal (Francis09). These structural styles may influence the terminology and word use in the document at the time it is produced which may not be known or relevant to the expert user in the future. There would be little evidence to the searcher to know the frame of mind of all the authors in a document corpus. An extreme example would be a modern layman user searching for a document written in early English prose such as Shakespeare.

Documents may mix these six writing styles in order to convey or record facts from the specific event it is describing. For example, a police report has both a chronological, categorical and causal structure. A police officer may record a confession of a suspect or the narrative of a witness, which includes quotes. These written records often contain slang or unorthodox sentence structure due to the need to quote exactly what a witness or suspect states to an officer. Such structural differences and nuances can cause missed searches of relevant documents because the search query fails to consider them. Something as simple as the distance between words can make the difference between a relevant phrase and two distinctly separate uses and meaning. A solution for this IR search problem might make use of structure to meet the requirement E of Table 1, Using Structure: Integration of Document Structure.

2.6 Query Expansion

Another challenge to the accuracy of searching in IR search systems is related to the search query itself. Because search terms are often only a few words or phrase, they are considered sparse. Sparse search terms do not contain enough information to allow a narrowing of the scope of what it is the searcher is looking to identify. Sparse queries generally have a large recall but poor precision. Another challenge is ranking the results in terms of relevance to the searcher when there is not enough information to determine relevance. Query expansion is a method used to try to provide more information for searches. Research has shown that enhancement of single term searches with a second term increases search precision dramatically (Whissel09). A popular method of query expansion is called relevance feedback (Rivas14). Relevance feedback systems often perform an initial query, then the searcher ranks the results in some way, grading the relevance of the returned items. Attributes of the selected items are used to add additional information to the original query term either implicitly or explicitly. When information is added it is called *information gain*. *Information Gain* is when new information is added to expand the understanding or the ability of a sparse piece if information to be more descriptive or accurate. The new enriched query is then executed against the corpus. An example of explicit feedback familiar to most people would be the auto-suggestion feature users see when they enter words into a Google search as they type. The tool is making ranked suggestions for the searcher because the more specific a query is made by expansion the more accurate the search results will be. Although this method of query expansion satisfies the search requirements for helping to make search easier for untrained users, it does not readily add to what the expert user would type on their own. This idea of *Information Gain* however, does satisfy the need shown in Table 1: not Just Ranked List: Consider Enriched Query Methods.

2.7 Lucene

Lucene is a popular open source full featured text search engine (Apache14). Lucene is a well-documented tool with a rich set of functionality useful for performing various forms of text analysis. Some of the most valuable tools available within Lucene includes the ability to create custom inverted indexes and the inclusion of the popular TF-IDF weighting system. There is a large community of support for this tool and it is extended and supported in new ways all the time. This makes it a good choice for use in document similarity and text analysis research (Hirsch10). Two prevalent tools in use today are built around the capabilities of Lucene because of its text searching and inverted index capabilities; SOLR and Elasticsearch. These tools provide developers with rich APIs to customize how data is built, stored, analyzed and distributed (Elastic14 and Solr14) Below is a process flow diagram describing how Lucene is typically used for text searches and analysis.

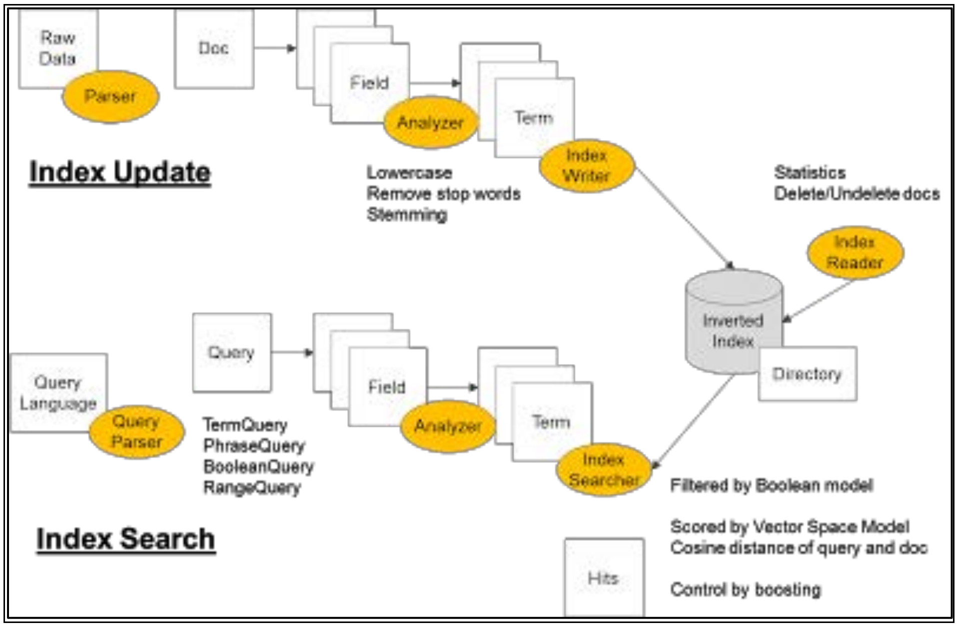


Figure 2: A process flow diagram of Lucene (Ho13).

First an index is created from the corpus of documents by using an analyzer which stems stop words, and tokenizes the text. Stemming stop words is removing or ignoring common words such as *a, and*, and *the*. Tokenization of text entails mapping terms to an identifier stored in a tree structure. The final step of creating the index is when the index writer constructs an inverted index, mapping the terms in the documents to their respective source documents. The second major function is to open an index and search it. The way Lucene conducts a search is using a binary search (analyzer) to find candidate documents, and then to score (searcher) those documents using a Lucene impleme ntation of the TF/IDF scorer shown in Equation 3. A ranked list is returned with document identification numbers and their respective TF/IDF scores in ascending order. This widely used search tool is very popular because of the ease with which it is implemented and how its scorer can be modified in an attempt to meet the requirements of this IR search problem.

2.8 Inverted Indexes

IR Search of documents makes use of indexing to perform searches on textual data. In order to avoid linear scans of texts during query operations, text documents are indexed in advance (Manning08). Documents are *tokenized*, an operation which breaks documents into individual terms. The *tokenized* terms are counted and stored in a tree structure with mappings back to their document. The structure resembles an inverted tree of indexes hence the name, inverted index. This enables very fast searches for individual search terms or words during a query (Whissel09). The open source tool called Lucene possesses the capability to create custom inverted indexes where mappings are created between words. Lucene indexing allows segmentation which allows for incremental addition of new documents. This allows for efficiency in not requiring rebuilding the index ass new documents are added over time (Apache14). Inverted indices satisfy the requirement for considering the information needs in a closed domain as well as to make the search tools easier for untrained users. Using an inverted index in Lucene removes the need to develop and manage a complex database system.

The nature of how data is managed and consumed for many IR search problems makes the use of a Lucene inverted index a popular choice.

Chapter 3

APPROACH

Previously, the use of TF/IDF scoring was shown as a popular way to rank document searches. It also suffers from too many false positive results. There is a need for new and improved methods for capturing and incorporating user context and written structure into user search queries. Restricted domains have information retrieval needs that are often unsupported by conventional tools.

In Chapter 2, various approaches and strategies were presented to improve various aspects related to IR searches: scoring, query expansion, structure and boosting/weighting. This research is based upon the assumption that a composite score, which includes a number of these strategies, will result in less false positives and more accurately capture the system user's unique perspective. These assumptions are listed in Table 3.

Table of Assumptions
1. A1: Composite scoring document searches reduces false positives
2. A2: Composite scoring document search results more closely resemble the user's perspective

Table 3: Table of Assumptions

The primary goal of this research is to improve the precision of ad-hoc document search rankings (results) over standard TF/IDF based ranked results by creating a Composite

Scorer, which reflects or directly uses these user perspectives to try to reduce the number of false positive results TF/IDF scoring currently suffers from. The approach of this research to solve the stated problem involves creating a software tool, which uses the Composite Scorer to produce accurate search results.

The approach is to meet the requirements listed in Table 1: (A) Consider Enriched Query Methods, (B) Make Search Easier For Untrained Users, (C) Incorporate User’s Context, (D) Consider information needs of restricted domains, (E) Integrate document structure. To meet those requirements a search tool must be employed, which allows multiple rank score algorithms, an easily integrated user interface, and the ability to easily access restricted data. The proposed solution needs to be built using a common computer language (C#) and be installed on a common platform (Microsoft Windows) in order to accommodate the needs of the FFC Crime Analysts. The following sections describe the proposed approach to computing rank scores.

3.1 Scoring Process

The approach to test the proposed solution is focused on how search scores are computed. Below is a process description of the approach to perform a scored ranking using two methods: baseline TF/IDF scoring and Composite scoring.

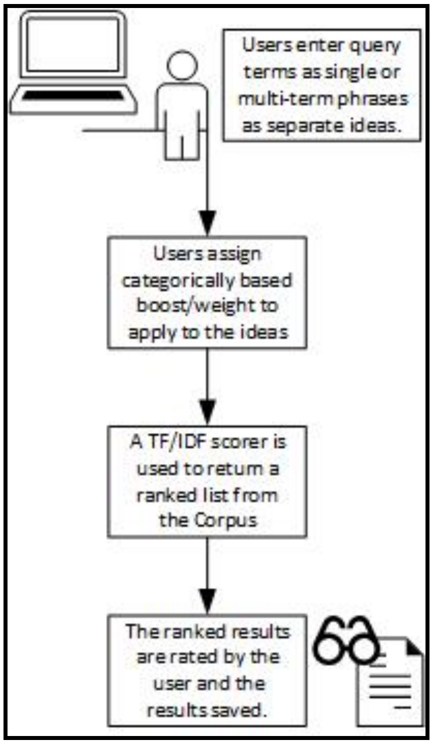

Users enter query terms as single or multi-term phrases as separate ideas.
Users assign categorically based boost/weight to apply to the ideas
A TF/IDF scorer is used to return a ranked list from the Corpus
The ranked results are rated by the user and the results saved.

Figure 3: Baseline Scoring Process

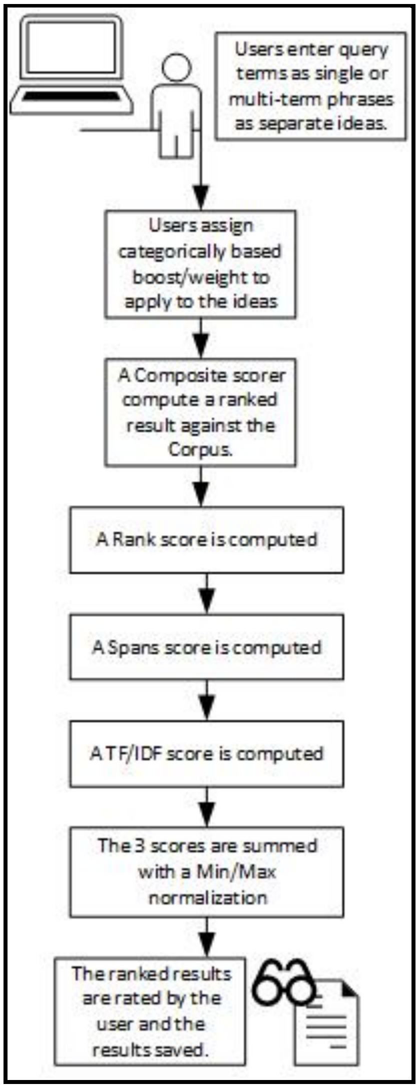

Users enter query terms as single or multi-term phrases as separate ideas.
Users assign categorically based boost/weight to apply to the ideas
A Composite scorer compute a ranked result against the Corpus.
A Rank score is computed
A Spans score is computed
A TF/IDF score is computed
The 3 scores are summed with a Min/Max normalization
The ranked results are rated by the user and the results saved.

Figure 4: Composite Scoring Process

3.1.1 Baseline Scoring

The baseline or standard scorer uses TF/IDF to compute the ranked score of documents when a searcher performs this type of search. When a user enters terms or multi-term phrases to construct a query, the scorer finds documents that contain at least one hit (occurrence) for at least one term of the user supplied query term(s). The TF/IDF scores for those documents are computed and the results are saved in a special object collection called a TopDocs object. This object stores the unique document identification number (ID) and the score for that document for this specific query. The ID is used to locate the terms in the index; it has no direct relationship to the content of the original document itself. The TopDocs object stores the “hits” for these documents in ranked descending order. The TF/IDF Scoring algorithm is described in the next section.

3.1.2 TF/IDF Score

Commercial and open source search tools use variations of the TF/IDF equation for computing vector scores for ranked search. An example is the Lucene TF/IDF scorer, which is a commonly used open source tool. The Lucene TF/IDF scoring function, shown below, produces a vector score for documents that contain the term or terms in a query. A term vector is a mathematical representation on a coordinate plane which represents a document and how close it compares to other document’s vectors. The TF/IDF scoring algorithm is shown in Equation 6.

$$score(q,d) = coord(q,d) \cdot queryNorm(q) \cdot \sum_{t\ in\ q} \left(tf(\ t\ in\ d) \cdot idf(t)^2 \cdot t.getBoost(\quad) \cdot norm(t,d)\right)$$

Equation 6: TF/IDF Score Formula (Apache14)

The Lucene TF/IDF Scorer shown includes a few methods specific to its implementation of the TF/IDF scoring equation. The first one is the coord(q,d) method. This method is an attempt to reward documents, which contain more of the query terms than a document that contains fewer terms. The queryNorm(q) function is a sum of squared weights normalizer which is used to make scores between queries comparable. For the composite scorer these normalizers are not used. The norm(t,d) function is used to retrieve information computed at document indexing. The index does not make use of these statistical computations during index building and so this function has no bearing in this approach. This was done to keep the implementation as agnostic as possible and decouple it from relying on the Lucene API. Weights are applied using the t.getBoost() function. It is used to apply boost at query time to documents with terms that the user has supplied a boost value for. These methods were neutralized for the testing of these methods assuming any TF/IDF scoring software would produce similar results, making the proposed approach as agnostic as possible. To illustrate how a TF/IDF search process may resemble, consider the following.

Given a search for the term “fox” in a corpus of 100 documents of which 50 of them contain at least one instance of the term “fox” a user would get 50 documents back from our search to score. For an example, one of those documents contains 20 other terms and the search term “fox” once. The TF/IDF calculation would be thus: TF = 1/20 or .05 and IDF = log(100 documents / 50 documents with the term “fox”) = .30102. Therefore, .05 * .30102 = .015051 is the TF/IDF score for this particular example document in this particular query from this particular corpus. TF/IDF scores always range between one and zero.

3.1.3 Composite Scoring

The proposed Composite Scoring approach uses three distinctly different scorers, which address differing aspects of the user-supplied query. In order to build a composite score the approach performs several distinct queries and then combines their scores. The queries chosen are called: Rank query, Span query, and TF/IDF query. Rank scores are computed at query time, which represent the number of user-supplied ideas contained in a document. The Rank query counts the occurrences for each idea or concept inside of each document in the corpus providing a ranked list of documents. The Span query provides a vector score ranked list of documents searching for any multi-term phrases. The score is computed on the number of terms depending upon the distance between them. This distance is determined by how much “phrase slop” the scorer is allowed. Phrase Slop refers to the number of unrelated terms allowed between desired terms in order to be considered a “hit” and counted in the scoring. The last scoring query is the TF/IDF query. The TF/IDF score is the least important to the computation of the composite score and is used to supplement the other two scores for each document. The scores are summed with the Rank query given precedence. The Composite Scorer makes use of a *Min-Max normalizer* (Equation 8) to ensure that the Rank Query has precedence. The Composite Scorer calculates a score based on three different measurements or aspects of the document and terms while also ensuring the most important aspect in this approach is the number of ideas in a document and not the frequency.

3.1.3.1 Rank Score

The Rank Score function simply returns a count of how many search or query terms appear in each document. The scorer emphasizes the number of ideas in a given document and not the frequency. The score for the query q of each document d equals 1 or 0 if the term t is found in the document. The formula is provided below.

$$\text{Score}(q, d) = \sum_{P(t)} f(t)$$

Equation 7: Rank Score Formula

The highest and lowest score is kept to use for Min-Max normalization during the calculation phase of the composite score when all the individual scores are summed.

3.1.3.2 Span Score

The Span Score function computes scores from multi-term phrases. It allows phrases of multiple terms to be found out of order based on the limits of the “phrase slop” setting. Phrase slop refers to the number of terms which can separate the search terms which make up a multi-term search. This allows any permutation of the terms which fall within the distance selected of one another will result in a “hit” for the query and be counted for scoring. The benefit of using this function is that typos or variations in how authors write documents is minimized and will allow for greater recall. The reason for its use is related to the structure of written documents. The distance between terms can dictate their relationship or semantic meaning to one another. As an example consider the following search term, “white truck”. A document with many sentences may contain the term “white” and the term “truck” but they may be several paragraphs away from one another, dramatically altering their relationship to one another.

3.1.3.3 Composite Score Normalizer

The Composite Scorer uses Min-Max normalization to prioritize the Rank Score above the other two scores. During experimentation, test users reported that the importance of the number of concepts in a document had more bearing in their investigations than simply frequency of use within a document. The Span and TF/IDF scores were secondary to the Rank score in importance according to early test results involved in the development of the approach. Min-Max normalization is often used in conditioning data during data-mining operations to ensure data fall within chosen ranges and to minimize outliers from affecting analysis. To ensure that Rank Score always had precedence in the ranking Min-Max normalization (shown in Equation 8) is applied.

$$Z_i = \frac{X_i - \min(X)}{\max(X) - \min(X)}$$

Equation 8: Min-Max Normalization

3.2 Categorical Boost

To achieve the goals of Table 1: Capturing Context: Incorporate User's Individual Context, the approach incorporates term boosting. Term boosting or weighting is when a number is selected as a multiplier to be used by a scorer in when computing scores. This allows the user to impart importance to a term, another user aspect. The default scoring is to make all terms weighted to a 1. This means that no term or phrase is more important than the other is from the user perspective. To meet another objective of Table 1: Help For Users: Develop Ways to Make IR Easier For Untrained Users, traditional boosting was augmented with the following options to convey user importance: Possible, Probable, Critical, Must, Never. The new approach provides that these values are configurable such that the actual weight values corresponding to categorical terms can be modified using a settings file through the graphical interface. For this research, the default settings chosen for the experiments are listed in Table 3.

Category	Weight Value
Possible	1
Probable	2
Critical	4
Must	6
Never	0

Table 4: Categorical Boost Default Settings

The categorical term identified as “possible” was set to the default weight of 1 since it may or may not be a part of the desired document. The “probable” term might be twice as important as a “possible” term so its setting was set at 2. A “critical” term might be twice as important as a “probable” one, so its weight was configured to a value of 4. Terms described as “Must” were weighted with a 6 to impart the importance to a user. Additionally, terms identified as “Must” limit the documents scored and returned to those containing the term or phrase at least once. The terms associated with the “Never” category have no weight value but the query is constructed to ensure that no documents with those terms are scored or returned at all. This categorical weighting showed that boost is applied to terms and phrases in a query using words that conveyed importance in a consistent way with users.

3.3 Score Assessment

In order to contrast or measure the performance between the baseline and composite scorer it is common to use both qualitative and quantitative measures (Manning08). Ranked documents by expert users will have qualitative aspects to their relevance, which varies from user to user. The expert users will have an important opinion on what is relevant and valuable and what is not. By measuring and contrasting both qualitative and quantitative results of the two scoring methods conclusions may be drawn as to the benefits of using a Composite Scoring approach.

To measure the results qualitatively, a simple cost function analysis can be used in order to reflect the importance of the possible results. The experts would perform queries on the corpus and then categorize the top ten documents for both methods. The categories are assigned a numerical cost based upon the cost of their impact or value to the analysts. As an example assume that, a document returned in a ranked list from a search scored lower than it should and was very important to finding a criminal. The cost of potentially missing this document because of its low score is relatively expensive. The categories were whether the analysts thought the document should have scored much higher, higher, just right, lower or much lower than they did. The measurement for quantitative analysis is “Precision at k” (Manning08). This method is widely accepted in IR measurement for top-k ranked results when what matters is how many good results are immediately available to the user.

To measure the quantitative performance of the Scorers in the experiment, Mean Average Precision (MAP) is used. Because ranked retrieval searches may return an indeterminate but potentially large number of ranked documents, a smaller selection is used to measure performance. For this research, the top ten documents of the ranked search results are analyzed for precision and recall. Several use cases are provided to the expert users with which to perform five searches. The analysts mark the top ten results and save their findings to a text file which saves the query terms and the rated results for both scoring methods for review. The precision and recall can then be calculated for each search query and an average mean can be computed with which to evaluate the average performance of both scoring methods. One of the benefits of using MAP to measure the performance of a search method is that a single numerical value is derived.

3.4 Conclusion

The preceding materials described the items and rationale used to form the experiment that follows. The TF/IDF Scoring method was reviewed to help explain how documents in a corpus are currently scored and ranked during a search query. The Composite Scorer presented in this research is explained and described as an approach to solving the stated problem. How the proposed solution can be measured is also explained. In the following chapters these items are assembled and tested.

Chapter 4

EXPERIMENTS

The previous chapters developed and explained the approach to meeting the requirements needed to improve the IR Search problem in a restricted domain using ad-hoc expert searches. In this chapter, the approach is realized in several experiments using a case study equipped with expert users. The objective of these experiments is to measure and evaluate the effectiveness of the proposed Composite Scorer approach contrasted to a traditional TF/IDF Scorer approach to solving the stated problems. The TF/IDF method in these experiments serves as the baseline method for search scoring. To evaluate the performance of the new approach, MAP and Cost measurements are contrasted to gauge whether an improvement is realized or not. The new approach is constructed to achieve the stated requirements of Table 1 as the objectives enumerated below.

Objectives
Accurately capturing/ describing the user’s unique concept.
Provide an easy to use interface.
Leverage inherent structure of multi-term phrases.
Increase precision over standard scoring methods.

Table 5: Experiment Objectives

To implement the Composite Score approach the Lucene (Apache14) API was chosen. It satisfied the needs for a readily modifiable API for custom and standard baseline scoring. It also allowed the development of the user application in C# allowing separation in the experiment between how it was implemented and the tools used to implement it. This assures that the approach can be realized with any number of technologies and is not dependent upon Lucene or any programming language specifically. Lucene also includes the tools to create inverted indexes saving time from developing a custom tool to accomplish this task. Lucene is a widely-used and known open-source API with the benefits of low cost and support for researchers and users alike.

Finding expert users in a restricted domain to perform any experiments can be difficult. The senior Crime Analyst from the Florida Fusion Center expressed needs for tools to overcome challenges presented previously in the Problem Statement. He was able to get permission for the agency to participate in this experiment as expert users. During the development of the user interface, the senior analyst acted as the chief stakeholder. This role was to facilitate the construction of the user interface. Fusion Centers are referred to as Investigative Support Centers (ISCs). These Centers act as both aggregators and disseminators of information between local, state, and federal agencies. They lawfully gather, analyze, and share information dealing with terrorism, crime, and public safety issues.

Because of the multiple agencies and jurisdictions involved, this restricted domain has many information regulations to meet. The nature of the work these agencies perform means a high variety of situations and thus many ad-hoc information needs. The size and amount of information that Fusion Centers must consume and analyze is very large and getting larger.

4.1 Test Case

The analysts at the Fusion Center struggle with performing analysis on up to a million new crime reports per month. Currently, the Fusion Center receives daily collections of reports from a multitude of law enforcement agencies across Florida. According to their expert analysts, current IR methods in use at the Center return too many false-positives and falsely ranked documents as being relevant. As a result, the analysts must manually examine large numbers of reports to ensure the concept of the expert investigator has been captured sufficiently, resulting in lack of confidence in the tools. Because of time constraints, analysts will only search a number of documents returned by the tools. Reportedly, current tools lack the ability to capture the concept of the expert users in an easy fashion allowing the analysts to adjust concepts using easy to understand language. These crime analysts often must perform multiple search queries in order to facilitate both Boolean searches and concept searches with complex SQL language queries. Consider the following example of a complex nested SQL Query.

```
SELECT Narrative, ReportID, ReportDate,
(SELECT *
FROM NCIS.DailyReports AS DailyReports
WHERE Date = date(‘11/01/2017’))
FROM NCISD1 as Report
```

Figure 5: Complex SQL Example

Another example would be where an analysts needs to search for documents containing both concepts of a crime such as “theft” and specific phrases such as “atm scanner”. The concept of “theft” may be far reaching while the specific documents containing “atm scanner” may happen to be mentioned in non-theft related police reports. There is no existing method where the analyst can easily create a search to express these or specific concepts from the analyst. Compounding the problem is an ever-growing input of data and the increasing number of investigations as populations in the jurisdictions of the agencies increase. The large number of text documents being uploaded to the Center daily requires more computing resources to process and store. A more efficient tool is needed to address costs in multiple human resources, opportunity costs of missed information searches and increase effectiveness of operations.

4.2 Experiment Development

The experiment involved developing and using an application with which expert analysts could perform side-by-side analysis using the baseline scoring and the new Composite Scoring Method. The following procedure was created to realize the experiment:

1. Create a simple and consistent user application with both the baseline and the new composite scoring capability.
2. Create a single corpus using existing FFC documents for use by both scoring methods.
3. Create use cases with simple instructions for the expert users.
4. Collect analysis results by expert users in researching the use cases.

5. Analyze and assess the results.

The experiment is a comparative analysis between standard TF/IDF scoring and the new composite scoring algorithm. The test used the same environment for both methods in order to ensure consistency and accuracy. The software application was used to create a corpus in the form of an inverted index. This index was used for the entire experiment to ensure consistency in the data source. Expert analysts in the FFC office were given the same use cases with which they constructed individual expert ad-hoc queries using the application interface. The analysts were tasked to implement two scoring methods, baseline and the composite scorer. The two methods were designated with ambiguous titles of “A” and “B” to act as a single blind technique hiding the scoring methods from the users to eliminate bias. Finally, the analysts were tasked to rate the results of both scoring methods recording their own analysis of the results. These results were collected and analyzed using the assessment methods described in chapters 5.1 and 5.2.

4.2.1 Creating the Application

One aspect of creating the expert system experiment is providing an easy to use interface. This is necessary because of the need to facilitate expert user analysis with ease and consistency. Multiple meetings with the stakeholders resulted in several findings. First, in order to facilitate building a search query they needed the ability to open a specific or known report, highlight elements within the document and create an expert search based on the selections. Second, the users needed a method whereby they could enter terms, phrases or other elements without the requirement of finding a known base document and still create an expert search composed of multiple terms or phrases. Third, users needed an easy way to emphasize the importance of different items or multiple items selected, deleted or entered ad-hoc. The fourth requirement was that the returned documents would need to be easily consumed and meaningful way. The fifth requirement addressed how the users would record their analysis of the returned documents and save those observations.

To solve the first requirement, a multiple document interface application (MDI) was programmed, to allow the ability to open multiple documents for review by an end user, much like the ubiquitous “Word” application by Microsoft. The application was built using C# which offered the ability to be easily implemented in the FFC environment. The main benefit of this design is the ability of the user to feel familiar with the interface. The more challenging question was how to capture or accept manually entered user information and construct a query while enabling the expert to emphasize their expertise.

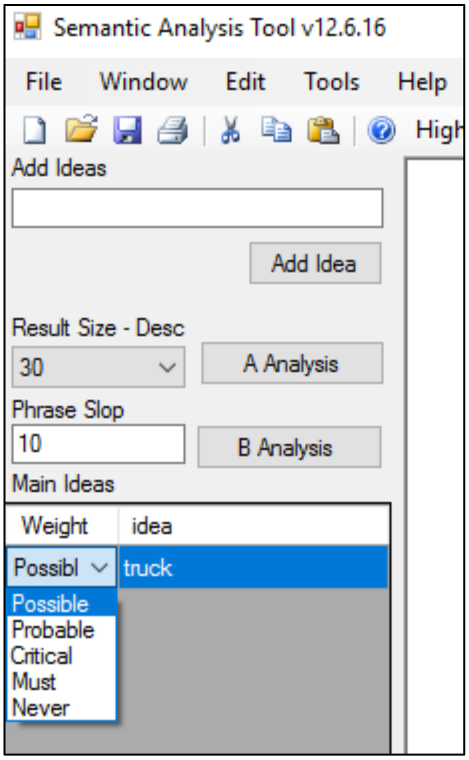
The interface was designed so that a user chooses terms and phrases based on what would be referred to as an idea. These ideas could be single or multiple term phrases and any ad-hoc query could contain any number of varying ideas which comprised a user concept. These ideas could be highlighted or found within a source document opened from the user interface or could be entered ad-hoc by the user without first finding a source document to act as a base.

A textbox entry method was used to help in satisfying the second requirement. If a user had one or more documents open in the reading pane of the application, the user could highlight terms or phrases and add them as ideas. Another way to enter in ideas was to type them into the textbox manually with no open document needed as a source. With this design, the user had the power to determine how to separate ideas which represent aspects of an overall concept. When a complete idea in the form of a single term, multi-part term or phrase was entered in the textbox, the user would “add” the idea to a collection of ideas that would be built as the user created or discovered new ideas to enter. This collection of ideas is referred to as the expert user’s concept.

The application now has a simple to use method to collect or create single word ideas, multi-term ideas or user created ideas using words and phrases. As an example, a user might be searching for drunk driver arrests in blue Chevy vans involving a white male. The searcher might select or enter a single term for “male” and a multi-term “blue Chevy van”. The application is built so that all text is converted to lower case in both the search tool and in the index comprising the corpus.

To solve the third requirement of applying semantic meaning to ideas, the application provides a method for users to add weights to ideas in the query. Weights impart semantic importance in queries when other information is absent. Traditionally, TF/IDF weights are simply numbers applied as multipliers to search scores of a given query. After discussions with the stakeholders, it was decided that numbers are ambiguous to a user. This would cause inconsistency between users based on training and fail to meet the requirement of an easy to use query tool. Therefore, the design allows the user to assign weights to the ideas in a more descriptive and meaningful manner by allowing for a language that made more sense to a user instead of asking them to enter a number. The categorical choices of: Possible, Probable, Critical, Must, and Never were chosen instead of numbers. The categories indicate the importance of an idea being in the results without needing any advanced query language training.

These categories and the numerical values they represent are modifiable in the application settings. The mechanism to configure the settings for all users of the application by a configuration file makes use of the extensible markup language (XML) format. The settings file can be edited or reconfigured by an administrator. The benefit of having the settings file is that changes can be made system wide in the application so that future work may more easily be facilitated. These settings are loaded when the application starts. A screen capture example of the implementation of the weighted ideas is shown in Figure 6 below.



Result Size - Desc: 30

Phrase Slop: 10

Weight	idea
Possibl	truck

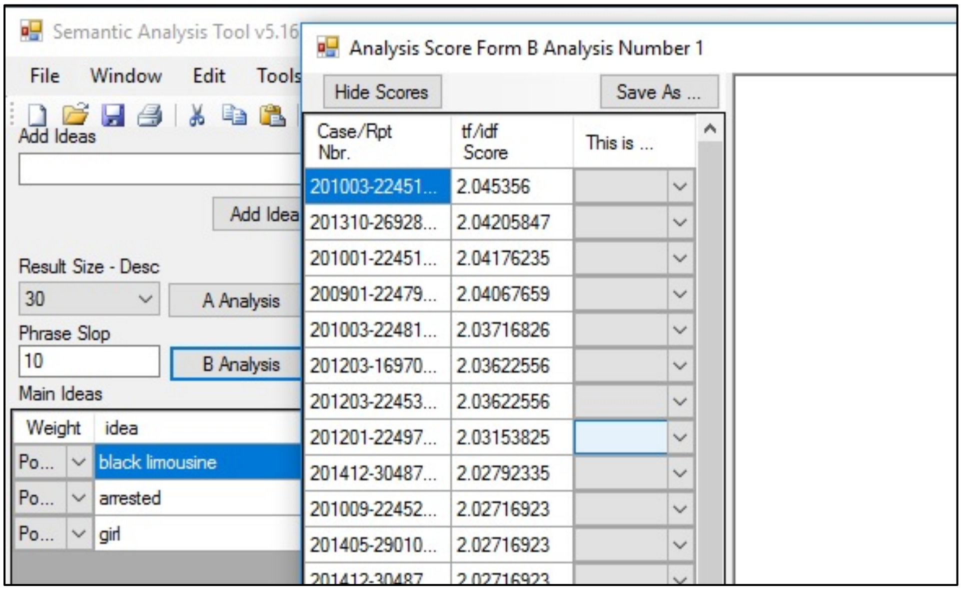
Figure 6: Expert User Categorical Weighting

Because the use case corpus contains several hundred thousands of indexed documents, the interface needs to allow the user to select how many documents to score and subsequently present in the returned ranked list. The application settings XML file contains default settings to populate a drop down menu named “Result Size”. For this experiment, the default was 30 documents. This meant that regardless of how many documents were found the user would only be presented with the top 30 scoring documents in descending score order. The expert analysts indicated that they rarely are able to analyze or study results sets larger than this using existing tools due to time constraints. This reemphasizes the importance of scoring the most relevant documents to the top of a ranked list regardless of the order of those top documents.

The final user selectable feature involved with applying user semantic meaning to ideas is named “phrase slop”. This effects how the Composite Scorer scores during the SpanQuery phase. It is not used in the TF/IDF scoring. This determines the maximum number of terms allowed between matching terms during scoring. The Composite Scoring Method uses a phrase query api call that allows the scorer to use the “phrase slop” value. The idea behind its use is that the closer terms are to each other from a multi-term phrase, the more likely the relationship to one another semantically. A large amount of phrase slop would result in higher retrieval but lower accuracy while a smaller might result in a missed concept or idea. This is influenced in the written structure of the documents as discussed in section 2.5 Structure.

The fourth requirement for the users involved is how to return the retrieved documents in a meaningful and useful way. As described previously, the document retrieval procedure involves a user creating a query or concept of ideas that the documents in the corpus are then scored against. The results would be a sorted list of scored documents when a query was performed regardless of the scoring method.

The Multi-Document Interface (MDI) design has an advantage by allowing each submitted query to return a new child-form (Result Form) containing a sorted list of scored documents. A child-form refers to a type of Windows Form that is created from a parent-form. The main application is the parent form in this case. The user can select a document by clicking on the document number shown in the list of scored documents of the Result Form. The sorted documents act as hyperlinks, which opens their source document in their entirety in a reading pane of the Result Form. A screen shot of the application and a Result Form are shown in Figure 6.



Result Size - Desc: 30

Phrase Slop: 10

Main Ideas

Weight	idea
Po...	black limousine
Po...	arrested
Po...	girl

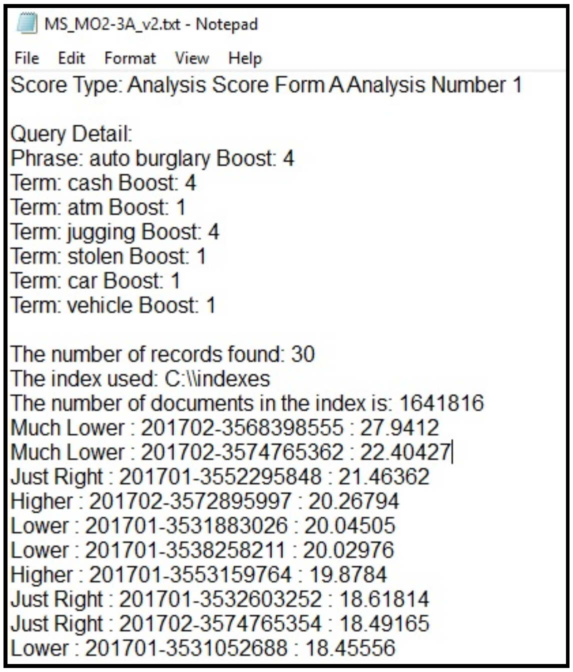
Case/Rpt Nbr.	tf/idf Score	This is ...
201003-22451...	2.045356	
201310-26928...	2.04205847	
201001-22451...	2.04176235	
200901-22479...	2.04067659	
201003-22481...	2.03716826	
201203-16970...	2.03622556	
201203-22453...	2.03622556	
201201-22497...	2.03153825	
201412-30487...	2.02792335	
201009-22452...	2.02716923	
201405-29010...	2.02716923	
201412-30487	2.02716923	

Figure 7: MDI Interface With Result Form

To make the user assessments easier, a highlighting scheme is applied on opened documents. This feature helps the analyst or user see where their ideas are located within the returned documents.

The fifth and final requirement for the user interface requires that the users are able to record their analysis of the search results and save it for future analysis and historical recording of the experiment. The population of data for this experiment consisted of just over a million plain text police reports from multiple police agencies. The data is only accessible by authorized personnel and any display to the public are required to be redacted. The identification of the analysts involved in the experiments has been obfuscated. To enable illustration of the experiments a way for the results to be saved capturing the query details, the unique id of each analyst, the internal document numbers, and the expert user's evaluations. The data recording scheme required consideration of these facts about the data and analysts involved in this research.

The Result Form is designed so that the user can select ratings for the chosen document from a pre-populated drop-down menu. Those choices were presented previously in section 3.6 Score Assessment. After the user has applied the ratings to the top 10 results, they save the Result Pane as a text file using the "Save As …" button on the Result Form. Several pieces of information are then saved to the text document preserving details of the query terms used, which scorer was used, the number of documents in the corpus at the time, any boost, the document number, and the computed score. Find an example file below in Figure 8.

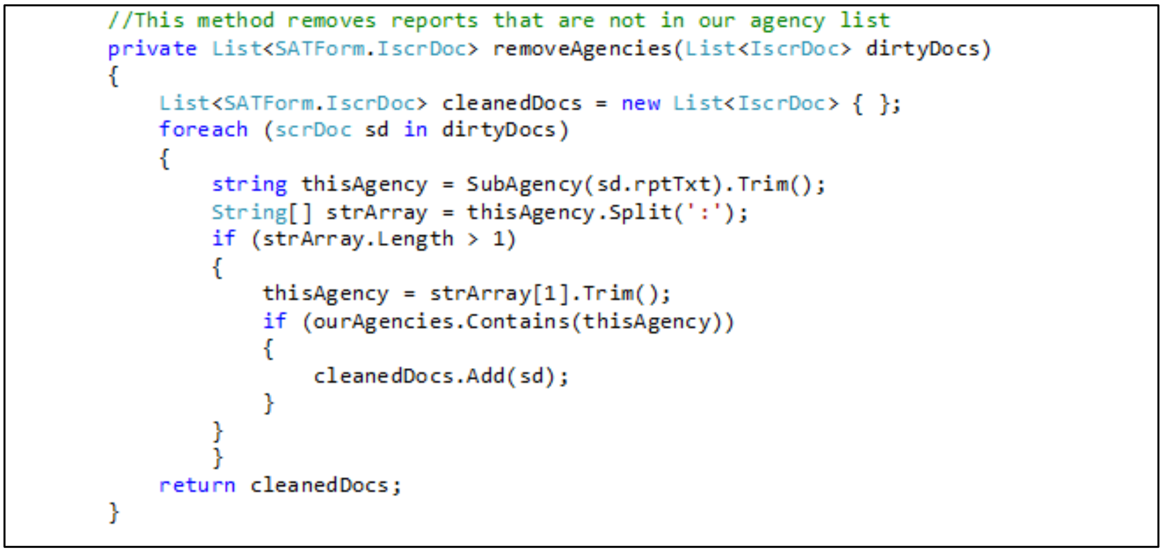


```
Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Phrase: auto burglary Boost: 4
Term: cash Boost: 4
Term: atm Boost: 1
Term: jugging Boost: 4
Term: stolen Boost: 1
Term: car Boost: 1
Term: vehicle Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Much Lower : 201702-3568398555 : 27.9412
Much Lower : 201702-3574765362 : 22.40427
Just Right : 201701-3552295848 : 21.46362
Higher : 201702-3572895997 : 20.26794
Lower : 201701-3531883026 : 20.04505
Lower : 201701-3538258211 : 20.02976
Higher : 201701-3553159764 : 19.8784
Just Right : 201701-3532603252 : 18.61814
Just Right : 201702-3574765354 : 18.49165
Lower : 201701-3531052688 : 18.45556
```

Figure 8: Example Results File

This would satisfy the final requirement for the user interface. This file allows for analysis of the performance of the scoring experiments.

During the development of the application, it was discovered that the corpus contained documents, which the analysts were not currently analyzing for various jurisdictional reasons. The Center analysts only monitor or analyze a number of North and Central Florida agencies. To allow the ability to analyze only certain agencies, a pruning method was created in the application. This would prune unwanted agencies from any search results. The XML settings file provides a setting named “Agencies” with an attribute named “clean” which may have a Boolean value indicating to prune or not prune agencies when the scoring query is performed. Figure 9 provides a snippet of the programming code which performs this operation.

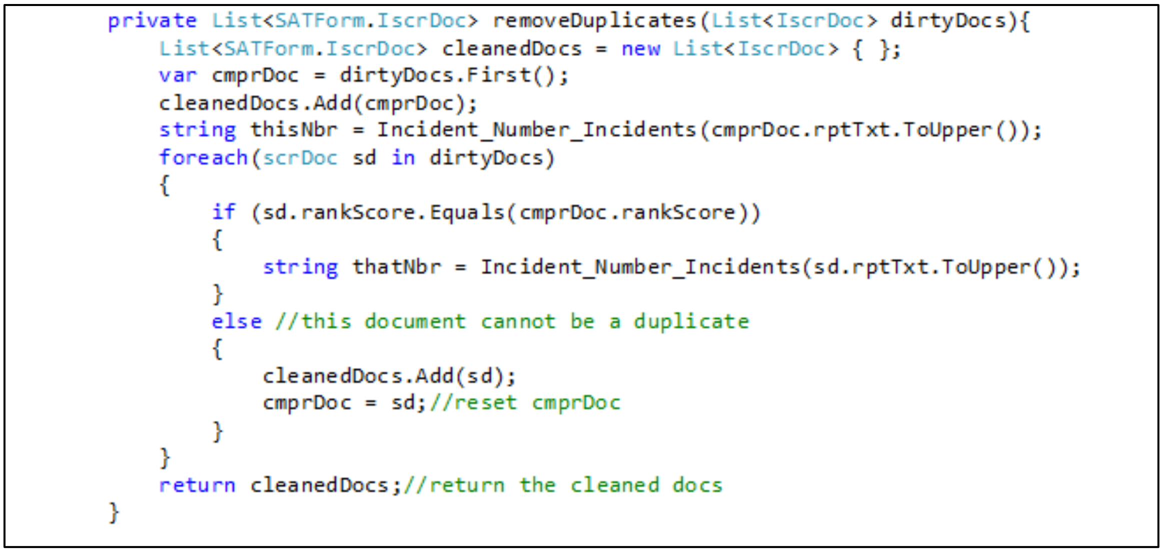


```
        //This method removes reports that are not in our agency list
        private List<SATForm.IscrDoc> removeAgencies(List<IscrDoc> dirtyDocs)
        {
            List<SATForm.IscrDoc> cleanedDocs = new List<IscrDoc> { };
            foreach (scrDoc sd in dirtyDocs)
            {
                string thisAgency = SubAgency(sd.rptTxt).Trim();
                String[] strArray = thisAgency.Split(':');
                if (strArray.Length > 1)
                {
                    thisAgency = strArray[1].Trim();
                    if (ourAgencies.Contains(thisAgency))
                    {
                        cleanedDocs.Add(sd);
                    }
                }
                }
            return cleanedDocs;
        }
```

Figure 9: Remove Unwanted Agencies

Another discovery during the design phase which concerned the index and corpus was that there were “duplicate” documents, which caused confusion to the analysts when they would review the results of any search. The document identifier is a unique number generated when they are delivered to the agency. Many police reports have addendum reports added to them over a period of time. These addendums rarely have critical information added to them and in this particular case they are most often used to change the status of a report and not the narrative when an investigation might go on over a period of time. As an example, a police officer may respond to a robbery and produce a police report. The police officer will conduct an investigation for several days in this example and conclude his investigative findings by submitting an addendum. These addendum reports cause duplicate narratives to be entered into the database and eventually into the corpus used in this research.

To mitigate the potential effect of duplicate reports in the search results a method was written in the the application. The method is enabled or disabled using the settings file by toggling the setting named “AllowDupes” with a boolean attribute “yes” or “no”. This allows the application to be configured to remove any duplicate documents from the returned ranked list. This is accomplished by looping through the returned ranked documents. The documents are compared and the scores are only stored once for any duplicate documents. The method is displayed in Figure 10.



```
private List<SATForm.IscrDoc> removeDuplicates(List<IscrDoc> dirtyDocs){
    List<SATForm.IscrDoc> cleanedDocs = new List<IscrDoc> { };
    var cmprDoc = dirtyDocs.First();
    cleanedDocs.Add(cmprDoc);
    string thisNbr = Incident_Number_Incidents(cmprDoc.rptTxt.ToUpper());
    foreach(scrDoc sd in dirtyDocs)
    {
        if (sd.rankScore.Equals(cmprDoc.rankScore))
        {
            string thatNbr = Incident_Number_Incidents(sd.rptTxt.ToUpper());
        }
        else //this document cannot be a duplicate
        {
            cleanedDocs.Add(sd);
            cmprDoc = sd;//reset cmprDoc
        }
    }
    return cleanedDocs;//return the cleaned docs
}
```

Figure 10: Remove Duplicate Documents

These two items about the pruning of agencies and the elimination of duplicate documents are included here to provide a sample of the programming as well as illustrate some of the challenges discovered during the experiments phase of this research.

4.2.2 Create a Single Corpus

An important step of the process is to create the corpus of documents from raw police reports. Documents are provided as groups of text files. These text files each represent a police report, a report addendum, or an investigative update to a report. Each report is associated with a Citizen Complaint Form (CCR) number. The police reports are tokenized, which breaks documents into single terms, and then added to an inverted index using the Lucene API.

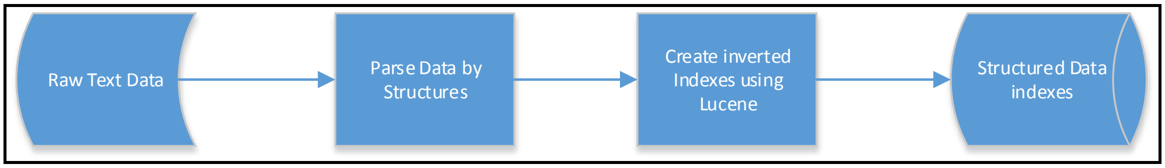


Figure 11: Creating inverted indexes from records.

The application needed to have data to query or score against. The case study users utilize a corpus of text-based police reports produced nightly. Reports from various agencies are received and placed into a folder structure for use by the investigators. The folders are segregated into months and years. These reports need to be in a format that the application can easily search and score. The documents were analyzed and stored in an inverted index. An inverted index is simply a hash table of the terms and the documents they map to.

The Lucene index building API provides tools which allow a high degree of customization in how indexes are built. In this test the simple analyzer was chosen. This allowed good speed by minimizing processing of the documents. Not wanting to build a dependency on the index builder to provide computations, the simplest analyzer was used. The Lucene “SimpleAnalyzer” takes text and breaks it up at non-letter characters in a process called tokenization. This causes marked up language and punctuation to be removed. The document is then stored into an inverted index. This satisfied the need to quickly access and make available existing data in the Fusion Center.

4.2.3 Create Test Use Cases

The lead analyst at the Fusion Center performing the test prepared several test use cases. These cases are scenarios and descriptions of crimes, motives or behaviors and ongoing investigations. The use cases provide a fixed set of needs for investigation for expert analysts to conduct individual search and analysis on. The use cases are separated by topic into the following general areas: Suspect Description (SD), Modus-operandi (MO), and Research Queries (RQ). The analysts were tasked with using the experimental application to perform a search and subsequently their expert analysis on any five of the available use cases. The instructions to the analysts were "*In this exercise, the below is all the information you have to go on. Use the provided tool to enter the words, individually or when near other words (like blue van or dog tattoo), in the search input lines provided. Use as many as you think reasonable to ensure you don't miss possible cases – ie catch things that may relate or be the hit you want before searching. Then assess the results of the top 10 per instructions in your video.*" The analysts performed their searches using both scoring methods and saved the results along with their ratings of each result. The use cases for the testing are detailed in Appendix A: Use Case Document. It should be noted that redactions were made were necessary to protect intelligence procedures of the agency as well as to protect sensitive information from public disclosure. The expert analysts were all given the use cases and instructions with which to perform the queries, tests, and evaluations. The evaluations saved by the individual analysts were then collected by the lead analysts and saved into a folder: Appendix B: User Analysis Data.

4.2.4 Collect Analyses Results

The application is constructed to allow the analysts to perform searches which use both TF/IDF scoring and the composite scoring. The analysts then rates the results one by one identifying documents as either being ranked well, too high, too low, or completely wrong. These correlate to positive, false positive, false negative, or negative. The details of the ratings, the returned documents, and the query terms are saved in a text file. The text files preserve the experiment and expert analysis for observation and measurement.

4.2.5 Analyze Results

The results saved previously by the analysts are collected and compiled into a single spreadsheet. The precision and recall are computed for each query to allow the MAP scores to be computed. The chosen costs are applied to the results and summed for the two scoring methods.

4.3 Conclusion

Chapter 4 explained how the solution to the problem and the proposed approach were realized. A large portion of the realization of the experiment involved programming a custom user interface for users to use in conducting the experiments. This application is still in use at the Center by the analysts and there has been indication of intention for its continued use and a desire for further refinements.

Chapter 5

RESULTS AND ANALYSIS

“The standard approach to information retrieval system evaluation revolves around the notion of *relevant* and *nonrelevant* documents.” (NLP14). This is traditionally measured using Precision and Recall. Another consideration in measuring the performance of a search method is that of cost. Cost can be a measure of the amount resources a search method consumes, the time a search method needs to perform, and the consequences of misclassified information. Some documents are more or less relevant than others are. These conditions are referred to as specificity and sensitivity. Precision measures the specificity of a search method while Recall is a measure of sensitivity. The assessment of the Composite Scorer and the TF/IDF scorer is an evaluation of ranked results. Therefore, precision and recall measures require extension from fixed sets of unordered documents to top k retrieved document sets (NLP14). For this research Mean Average Precision of the first or “top” ten results of a search will be calculated.

Other measures of performance are more subjective and often specific to the information domain being searched. An illustration of this would be an information system that is used to gather evidence that could either exonerate or condemn a suspect involved in a crime. Accepting that a perfect accuracy rate is unattainable, even a highly accurate precision rate may not be acceptable in some cases. As an example, assume that a search failed to identify or score a document as relevant and the user never received it. That one missed document may contain important information, which could exonerate a suspect of a crime. Missing this document in this example condemns the suspect to prison for a crime he did not commit. A Cost measurement is calculated in this research to contrast the cost benefit or detriment between the two scoring methods. This cost measurement took into consideration how a result may be more costly than others.

5.1 Cost Analysis

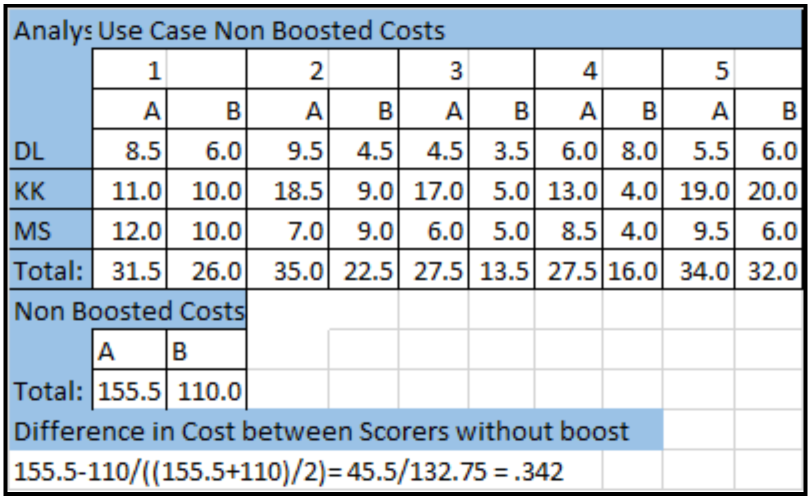
The first measurement applied to the result set is a simple cost computation. There are five categorical values assigned by the expert analysts. These are assigned by the individual analysts (Users) based on their evaluation as to how the ranking of the top ten documents are scored. Depending upon the ranking they choose we assigned a penalty to compute a cost. The first rating is “Much Higher”. This means the document should have scored higher than it did and has a higher chance of being missed by an analyst even though there may be high recall. Because of this, the cost for this rating is the most expensive at two. The next rating is “Higher” with a value of one. The next rating is “Just Right” which means it is a perfect find with a cost of zero. A rating of “Lower” means the document scored higher than it should have but was better to ensure it was seen than not seen since it is a relevant document. This rating has the smallest penalty of .5. The final rating is “Much Lower” and like “Lower” it is not very expensive to the researcher to have a document score higher than it should and has a cost of one.

To calculate the percentage difference in cost between the two scoring methods we apply the formula in Equation 9.

$$percent\ difference = \frac{difference\ of\ values}{average\ (mean) of\ values}$$

Equation 9: Percent Difference of Cost

The first experiment involved three analysts and five use cases. The analysts used the software tool to construct non-boosted searches. Below in Table 6 are the results after the cost function has been applied and the calculations for the cost difference between scoring methods A and B, TF/IDF and Composite Scorer respectively.

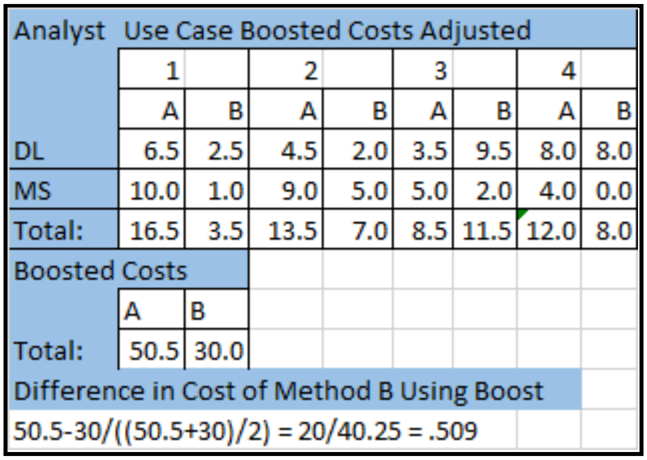


Analys	Use Case Non Boosted Costs									
	1		2		3		4		5	
	A	B	A	B	A	B	A	B	A	B
DL	8.5	6.0	9.5	4.5	4.5	3.5	6.0	8.0	5.5	6.0
KK	11.0	10.0	18.5	9.0	17.0	5.0	13.0	4.0	19.0	20.0
MS	12.0	10.0	7.0	9.0	6.0	5.0	8.5	4.0	9.5	6.0
Total:	31.5	26.0	35.0	22.5	27.5	13.5	27.5	16.0	34.0	32.0
Non Boosted Costs										
	A	B								
Total:	155.5	110.0								
Difference in Cost between Scorers without boost										
155.5-110/((155.5+110)/2)= 45.5/132.75 = .342										

Table 6: Cost Analysis Table 1

The analysts are listed under the “Users” columns and the use cases have a cost computation for each scoring function A and B. The total cost summation for Test 1 method “A” is 155.5 and for “B” its 110. When the cost calculations are computed using the formula, a measurement of 34% less costly results using the composite score over the baseline score when no user weights are used.

The second table shows the costs of two of the analysts on four use cases with the same search terms but making use of boost to impart importance and meaning. The difference in the number of use cases was because one of the analyst missed applying boost on the last use case using method A rendering that data unusable. There were also only two analysts available to perform the second experiment. The Table 7 below shows adjusted data tables to fit the data for analysis.

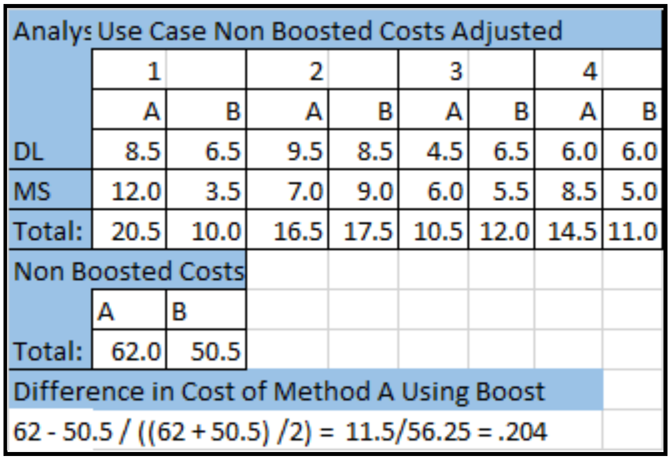


Analyst	Use Case Boosted Costs Adjusted							
	1		2		3		4	
	A	B	A	B	A	B	A	B
DL	6.5	2.5	4.5	2.0	3.5	9.5	8.0	8.0
MS	10.0	1.0	9.0	5.0	5.0	2.0	4.0	0.0
Total:	16.5	3.5	13.5	7.0	8.5	11.5	12.0	8.0
Boosted Costs								
	A	B						
Total:	50.5	30.0						
Difference in Cost of Method B Using Boost								
50.5-30/((50.5+30)/2) = 20/40.25 = .509								

Table 7: Cost Analysis Table 2

The total cost summation for Test 2 method “A” is 51 and for “B” its 30. The small set of numbers is because there were not as many analysts available for the test within the time constraints. Additionally, one use case test was incomplete and so it was left out to minimize bias. The calculated cost difference between the two methods when weights are applied shows approximately a 51% reduction in cost in favor of the Composite Scorer using boost.

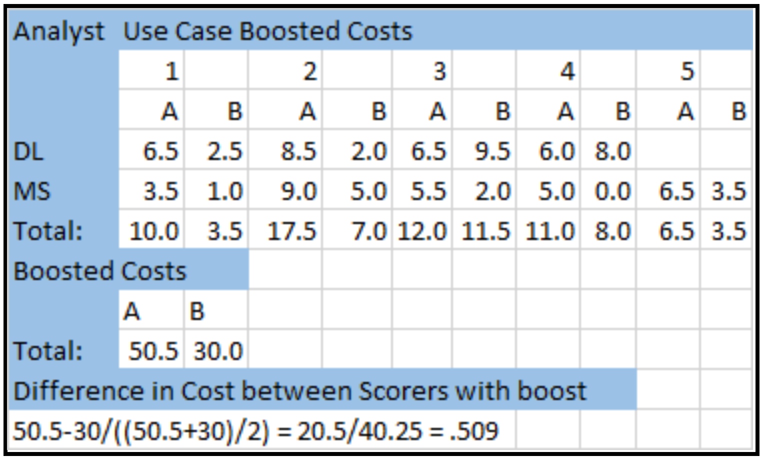
The third table in Table 8 represents cost calculations of scoring method A without boost versus scoring method A with boost.



Analys	Use Case Non Boosted Costs Adjusted							
	1		2		3		4	
	A	B	A	B	A	B	A	B
DL	8.5	6.5	9.5	8.5	4.5	6.5	6.0	6.0
MS	12.0	3.5	7.0	9.0	6.0	5.5	8.5	5.0
Total:	20.5	10.0	16.5	17.5	10.5	12.0	14.5	11.0
Non Boosted Costs								
	A	B						
Total:	62.0	50.5						
Difference in Cost of Method A Using Boost								
62 - 50.5 / ((62 + 50.5) /2) = 11.5/56.25 = .204								

Table 8: Cost Difference Table 3

When TF/IDF or score method A was used with boost, the cost was reduced by 20% as compared to simply using no term weighting with TF/IDF scoring. Method A and B cost scores are shown in the Table 9.



Analyst	Use Case Boosted Costs									
	1		2		3		4		5	
	A	B	A	B	A	B	A	B	A	B
DL	6.5	2.5	8.5	2.0	6.5	9.5	6.0	8.0		
MS	3.5	1.0	9.0	5.0	5.5	2.0	5.0	0.0	6.5	3.5
Total:	10.0	3.5	17.5	7.0	12.0	11.5	11.0	8.0	6.5	3.5
Boosted Costs										
	A	B								
Total:	50.5	30.0								
Difference in Cost between Scorers with boost										
50.5-30/((50.5+30)/2) = 20.5/40.25 = .509										

Table 9: Cost Difference Table 4

The composite scorer benefited even more from the addition of boost with a reduction in costs of 51%. The Composite Scorer outperformed the TF/IDF Scorer, reducing the potential cost of failed searches by half. Table 10 shows the summation of costs for both scorers.

Search	Cost	% difference
TF/IDF (Method A) with no boost	155.5	34%
CMS (Method B) with no boost	110	
TF/IDF (Method A) with boost	50.5	51%
CMS (Method B) with boost	30	

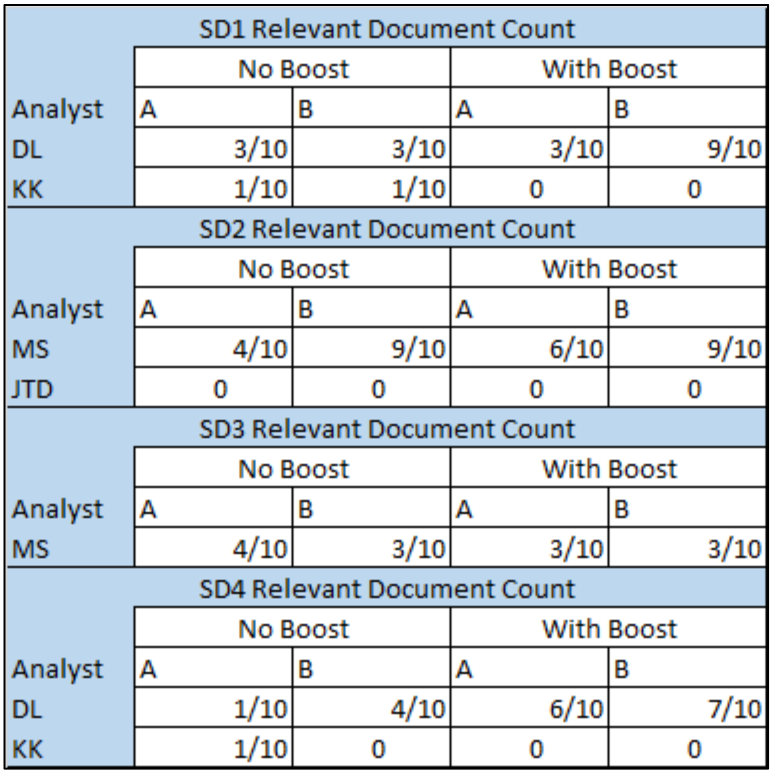
Table 10: Cost Summary

5.1.2 Precision & Recall

Precision and recall for each complete data set is found in Appendix C. A complete data set has the top ten documents for all four scoring methods; Method A with no boost, Method A with boost, Method B with no boost and Method B with boost. The precision and recall for each search is calculated and then the Mean Average Precision (MAP) of each of the four Methods is computed. MAP is a common single-figure measure of quality when a fixed number of top k (10 in this case) documents are retrieved and measured. MAP approximates the area under a precision-recall curve and is easy to understand because it calculates to a single numeric value (NLP14).

The first step in calculating the precision and recall requires the analyst to evaluate how many correct “hits” or relevant documents are returned with each scoring method for each of the selected use cases. These “hits” are graded on whether or not the returned documents are identified as relevant by the analyst performing the search.

The first group of use cases are titled Suspect Description (SD). These mimic one of three common use case scenarios the analysts normally conduct while searching for documents. These SD use cases often reveal information related to crimes involving suspects that matched a given description. The hit results of the SD searches are shown in Table 11.



	SD1 Relevant Document Count			
	No Boost		With Boost	
Analyst	A	B	A	B
DL	3/10	3/10	3/10	9/10
KK	1/10	1/10	0	0

	SD2 Relevant Document Count			
	No Boost		With Boost	
Analyst	A	B	A	B
MS	4/10	9/10	6/10	9/10
JTD	0	0	0	0

	SD3 Relevant Document Count			
	No Boost		With Boost	
Analyst	A	B	A	B
MS	4/10	3/10	3/10	3/10

	SD4 Relevant Document Count			
	No Boost		With Boost	
Analyst	A	B	A	B
DL	1/10	4/10	6/10	7/10
KK	1/10	0	0	0

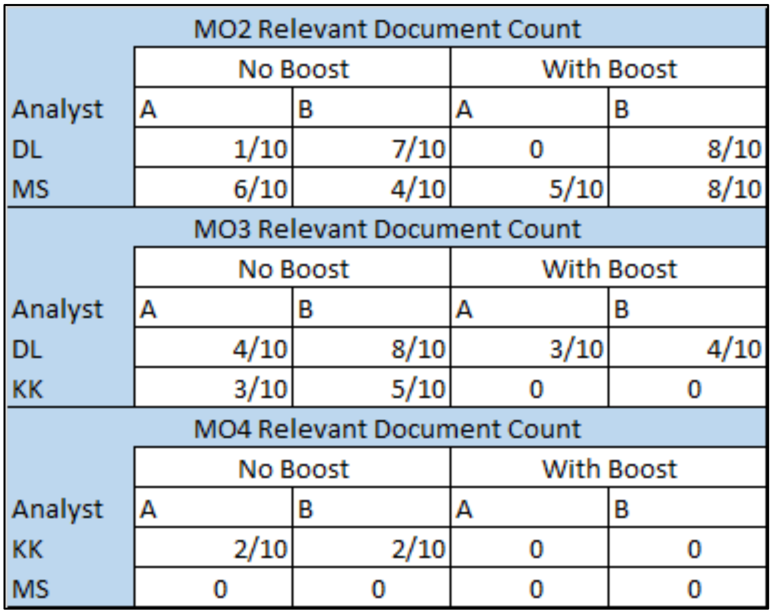
Table 11: SD Use Case Hit Results

For use case SD1 analysts were tasked to search for documents about a black male with sleeve tattoos on both arms driving a black or dark colored Impala. The results for one of the analysts conducting the experiment is missing or incomplete and so there is only one set of test results from one analyst in this SD1 use case experiment. The most interesting observation from this table is how the searches without boost performed equally accurately for method A. DL used nine different multi-term phrases in his query. In contrast, KK chose two multi-term phrases and two single terms. When DL applied boost to his terms, this analyst placed emphasis on four multi-term phrases. A possible explanation for the relatively worse performance of method A when compared to method B for boosted search might be explained by the effect of the Rank Score allowing documents with more terms in them to be better represented in the results.

The experiment on use case SD2 involved searching for a white male with red hair and a beard last seen riding a blue Harley Davidson motorcycle. One of the analysts failed to perform the composite scoring query so there is only one set of results shown.
The results of this experiment are inconclusive as there is a severe lack of data.
The experiment on SD3 involved searching for a Hispanic female with a thin white male accomplice in shoplifting. Only one set of data was recorded for this particular use case as only one analyst chose perform a search using this use case.

The experiment for SD4 involved searching for a white male KKK member or Neo Nazi with an alias of Ryan. This experiment was missing data from one of the analysts. The one good set of data reflected that the analyst identified as DL favored the results returned by the composite scorer. For the SD Use Cases the Composite Scorer (Method B) performed better than the TF/IDF Scorer (Method A) did for both boosted and non-boosted searches.

The next use cases involved the Modus Operandi (MO) searches. These are scenarios where the analysts are asked to search for documents which fit a pattern for crimes. The results for these experiments are shown in Table 12.



	MO2 Relevant Document Count			
	No Boost		With Boost	
Analyst	A	B	A	B
DL	1/10	7/10	0	8/10
MS	6/10	4/10	5/10	8/10
	MO3 Relevant Document Count			
	No Boost		With Boost	
Analyst	A	B	A	B
DL	4/10	8/10	3/10	4/10
KK	3/10	5/10	0	0
	MO4 Relevant Document Count			
	No Boost		With Boost	
Analyst	A	B	A	B
KK	2/10	2/10	0	0
MS	0	0	0	0

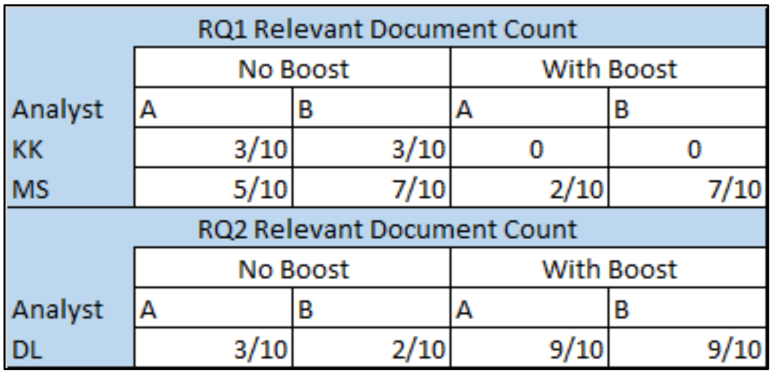
Table 12: MO Use Case Hit Results

Experiment MO1 involved a search for documents where firearms had been stolen during burglaries and also where the suspect had disabled the security systems. None of the analysts selected to investigate this use case, which explains its absence.

Then next use case was MO2, which involved searching for cases about victims who had cash in their vehicles that were then subsequently stolen. This is referred to in slang as “jugging”. Unfortunately, this data set is incomplete but based upon the overall table it is easy to infer that the missing data point for this use case would reflect a similar pattern of seen for Method A with no boost.

Experiment MO3 involved searching for cases where burglary suspects entered through the roofs of businesses. Even with the missing data the composite scorer shows improvement over the TF/IDF scorer in this use case. Interestingly, the hits decreased for both methods when boosting was applied. Inspection of the actual query for analyst DL revealed boost was selected for two very specific phrases, “roof entry” and “business burglary”. There is a lack of evidence to draw a conclusion from this observation.

The final use cases involved Research Queries (QR) cases whereby the analysts are asked to research for cases involving various topical elements such as homelessness and abandoned buildings. The results are listed in Table 13.



RQ1 Relevant Document Count				
	No Boost		With Boost	
Analyst	A	B	A	B
KK	3/10	3/10	0	0
MS	5/10	7/10	2/10	7/10

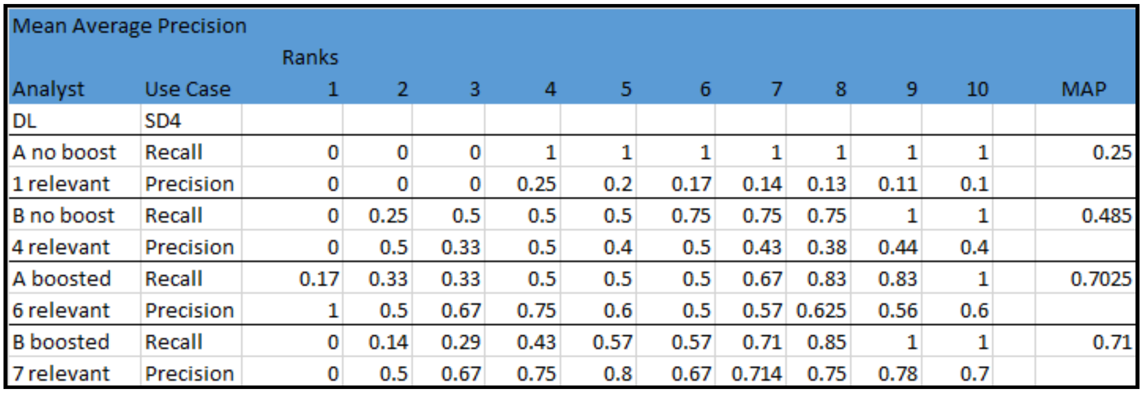
RQ2 Relevant Document Count				
	No Boost		With Boost	
Analyst	A	B	A	B
DL	3/10	2/10	9/10	9/10

Table 13: RQ Use Case Hit Results

The RQ1 experiment involved researching homeless people in abandoned buildings. The scoring for this use case appears evenly matched when no boosting is chosen for either scoring method. Further investigation on how the analysts constructed their queries showed that there was exceptional sparsity of terms chosen. This may explain the similar performance between the two scorers.

The RQ2 experiment also reflected similar scores between the two methods but with higher ratings using boost. By examining the actual construction of the queries, it was revealed that the analyst had applied boost to each of the search terms equally. This helps explain why the scores were similar between methods with no clear winning scorer for this experiment.

The precision and recall for each of the experiments was then calculated. These calculations were then used to compute a MAP score for both each method for each experiment but also MAP scores for all four scoring methods collectively. As an example, Table 14 summarizes the precision and recall for use case SD1.



Mean Average Precision													
		Ranks											
Analyst	Use Case	1	2	3	4	5	6	7	8	9	10		MAP
DL	SD4												
A no boost	Recall	0	0	0	1	1	1	1	1	1	1		0.25
1 relevant	Precision	0	0	0	0.25	0.2	0.17	0.14	0.13	0.11	0.1		
B no boost	Recall	0	0.25	0.5	0.5	0.5	0.75	0.75	0.75	1	1		0.485
4 relevant	Precision	0	0.5	0.33	0.5	0.4	0.5	0.43	0.38	0.44	0.4		
A boosted	Recall	0.17	0.33	0.33	0.5	0.5	0.5	0.67	0.83	0.83	1		0.7025
6 relevant	Precision	1	0.5	0.67	0.75	0.6	0.5	0.57	0.625	0.56	0.6		
B boosted	Recall	0	0.14	0.29	0.43	0.57	0.57	0.71	0.85	1	1		0.71
7 relevant	Precision	0	0.5	0.67	0.75	0.8	0.67	0.714	0.75	0.78	0.7		

Table 14: Use Case SD1 Precision and Recall Chart

What these charts display is how the precision changes as the recall increases. Precision is a measurement of the probability that a retrieved document is relevant. Recall is the probability that all relevant documents will be retrieved. Typically, the relationship between the two is inversely proportional. MAP provides an average overall measure of the performance for a Top(k) query under the Precision and Recall curve, which makes it a good indicator of the overall performance of a query.

Chapter 6

CONCLUSIONS

The experiments conducted in this research support the suggested approach of using a Composite Scoring Method for making use of user perspective and leveraging written structure. The results of the experiments in this test case improved the accuracy and effectiveness of traditional term frequency scoring methods. Creating a composite score, which includes consideration of the structure of user concepts and ideas, the frequency, and the number of ideas found in documents, provides significantly improved expert system document retrievals. The empirical evidence collected from this experiment demonstrates that Composite Scoring Methods outperforms traditional TF/IDF Scoring.

6.1 Approach Effectiveness

The Composite Scoring Method (CSM) shows it is superior to the TF/IDF Scoring Method in this test case. The summative characteristics of the Composite Scorer enable multiple aspects of a document or information need to be identified and a score calculated. Adding more scoring data functioned as a source of Information Gain for the overall CSM Score, which effectively added information to the overall query and thus improved the score of relevant documents. The Composite Scoring Method did not require any extra input from the users or external sources. CSM was able to produce better search results with the same sparse data input from users than TF/IDF.

6.2 Composite Scoring Without Boost

Despite variations between the analysts interpretations and opinions on a shared data corpus, the overall categorical observation of the composite score performance over the baseline method was significant at a 34% less costly performance. While this percentage may not correlate to a direct 34% increase in production, it can be reasoned that the improvements to both productivity and the quality of that production would increase significantly. With empirical evidence, confidence in the tools can also increase the use of tools by users. Having confidence in the tools will encourage its use and affect the overall quality of the work done by the analysts.

The MAP scores demonstrate that the Composite Scoring Method offers increases in overall accuracy of searches in both boosted and non-boosted searches over TF/IDF. Figure 11 shows that accuracy for CSM (Method B no boost) is increased by approximately 12% in non-boosted queries over TF/IDF (Method A no boost).

Search Method	MAP Score
TF/IDF (Method A) with no boost	50.46%
TF/IDF (Method A) with boost	46.85%
Composite (Method B) with no boost	62.17%
Composite (Method B) with boost	69.66%

Figure 12: Composite MAP Scores

6.3 Composite Scoring With Boosting

When composite scoring was combined with weighting or term boosting, the results on costs were even more dramatic. A 51% reduction in costs as compared to the TF/IDF scoring method was observed (Table 8) using the cost function methodology. The MAP Scores demonstrated that the accuracy was improved by 23% between using boosted TF/IDF (46.85%) and boosted composite scoring (9.66%).

6.4 Additional Work

The Composite Scoring Method offers the flexibility to be expanded to include other informational aspects such as incorporating the use of an ontological database. The ability to provide a weight to aspects of different pieces of the Composite Scorer also offers the opportunity to easily introduce other methodologies into the scorer, which for example might use Artificial intelligence (AI), Machine Learning or other supporting IR systems.

Another need uncovered during this research was adapting to individual user’s search needs. Each user of an IR search tool is going to be different in how they interact with the tool. A tool, which can learn from the user based on their history of usage and how they rated the results, could enable an IR search tool with much higher accuracy in precision for each user over time.

6.5 Summation

Using the Composite Scoring Method increased accuracy while reducing costs for expert ad-hoc queries. The application tool that was developed enabled users to apply their unique perspective to searches with a minimum of training or instruction. The growing amount of documents and information involved in many closed domains, such as the intelligence and law-enforcement community, reflects the need for improved IR systems. The use of composite scoring for these IR search systems will enable quicker and more accurate response over traditional scoring methods to assist these agencies in dealing with present and future needs.

REFERENCES

Print Publications

[Allan12]
Allan, J., B. Croft, A. Moffat, M. Sanderson, “Frontiers, Challenges, and Opportunities for Information Retrieval,” Report from SWIRL ACM SIGIR Forum Vol 46(1), June 2012, pp. 2-32.

[Belkin08]
Belkin, N., “Some(what) grand Challenges for Information Retrieval,” ACM SIGIR Vol 42(1), June 2008, pp. 47-54.

[Bezzazi07]
Bezzazi, E. H., ”Building An Ontology That Helps Identify Criminal Law Articles That Apply To A Cybercrime,” ICSOFT 2007, Proceedings of the Second International Conference on Software and Data Technologies, Volume PL/DPS/KE/MUSE, Barcelona, Spain, July 2007.

[Brinkmeir06]
Brinkmeier, M., “Pagerank Revisited,” ACM Transactions on Internet Technology (TOIT), Vol 6(3), August 2006, pp. 282-301.

[Chaplot14]
Chaplot, D. S., P. Bhattacharyya, “Literature Survey on Unsupervised Word Sense Disambiguation,” A thesis from The Department of Computer Science and Engineering Indian institute of technology, Mumbai, India, May 2014.

[Croft09]
Kim, J., W.B. Croft, “Retrieval Experiments using Pseudo-Desktop Collections” Conference on Information and Knowledge Management, Hong Kong China (CIKM’09), November 2009, pp. 1297.

[Elsayed08]
Elsayed, T. and Lin, J. and Oard, D. W., “Pairwise Document Similarity in Large Collections with MapReduce,” Proceedings of ACL08: HLT, Short Papers, Association of Computational Linguistics, Columbus, Ohio, June 2008, pp. 265-268.

[Erk08]
Erk, K. and Padò, S., “A Structured Vector Space Model for Word Meaning in Context,” Proceedings of the 2008 Conference On Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Honolulu, HI, October 2008, pp. 897-906.

[Grefenstette09]
Grefenstette, E., and S. Pulman. "MSc Computer Science Dissertation Analysing Document Similarity Measures." A Thesis from the University of Oxford, UK. November 2010.

[Hirsch10]
Hirsch, L., “Evolved Apache Lucene SpanFirst Queries are Good Text Classifiers,” 2010 IEEE Conference on Evolutionary Computation (CEC), Barcelona, Spain, July 2010, pp. 1-8.

[Han12]
Han, J., M. Kamber, J. Pei, “Data Mining Concepts and Techniques,” Morgan Kaufmann Publishers, Waltham, Massachusetts, 2012.

[Khan14]
Khan A. Javed, “Comparative Study of Information Retrieval Models Used in Search Engines,”IEEE Conference on Advances in Engineering & Technology Research (ICAETR - 2014), IEEE, Unnao, India, August 2014.

[Landauer98]
Landauer, T.K. and Foltz, P.W. and Laham, D., “An Introduction to Latent Semantic Analysis,” Discourse Processes, Vol 25(2), 1998, pp. 259-284.

[Manning08]
Manning, C., Prabhakar, R., Hinrich, S., “Introduction to Information Retrieval,” Cambridge University Press, New York, NY, July 2008.

[Rivas14]
Rivas, A.R., E. L., Iglesias, L., Borrajo, “Study of Query Expansion Techniques and Their Application in the Biomedical Information Retrieval,” The Scientific World Journal, Vol 2014, Article ID 132158, 2014.

[Sanderson12]
Sanderson M., W. B. Croft, “The History of Information Retrieval Research,” Proceedings of the IEEE, Vol 100, Special Centennial Issue, 2012, pp.1444-1451.

[Selvi14]
Selvi R. Thamarai, George E. Dharma, “An Approach to Improve Precision and Recall for Ad-hoc Information Retrieval using SBIR Algorithm,” World Congress on Computing and Communication Technologies, WCCCT 2014, Tiruchchirappalli, India February 2014. pp. 137-141.

[Sahlgren05]
Sahlgren, M., “An Introduction to Random Indexing,” Swedish Institute of Computer Science, Kista, Sweden, 2005, pp. 1-9.

[Shen13]
Shen, W., J. Wang, J. Han, “Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions,” IEEE Transactions on Knowledge and Data Engineering, Vol 27(2), May 2014, pp. 443-460.

[Stanchev12]
Stanchev L., “Building Semantic Corpus from WordNet,” 2012 IEEE international Conference on Bioinformatics and Biomedicine Workshops, Philadelphia, PA, October 2012, pp. 226-231.

[Stokoe03]
Stokoe, C., M. P. Oakes, J. Tait,”Word Sense Disambiguation in Information Retrieval,” ACM SIGIR’03, Toronto, Canada, July 2003.

[Weiling14]
Weiling, C., G. Wang, FengXia, Y., “Document Similarity Calculations Model of CSLN,” 2014 IEEE International Conference on Software Engineering and Service Science, Beijing, China, June 2014.

[Whissel09]
Whissel J., “Information Retrieval Using Lucene And Wordnet,” A Thesis from the University of Akron, Akron, OH, December 2009.

[Zhong12]
Zhong, Z., H.T. Ng, “Word Sense Disambiguation Improves Information Retrieval,” Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, Jeju, Republic of Korea, July 2012, pp. 273–282.

[Zhuge11]
H. Zhuge and J. Zhang, “Automatically constructing semantic link network on documents,” Concurrency and Computation: Practice and Experience, Vol 23(9), June 2011, pp. 956-971.

Electronic Sources

[Apache14]
http://lucene.apache.org/core/3_6_2/api/core/org/apache/lucene/search/Similarity.html (Aug 2017)

[Chen17]
M. Chen, “TF-IDF, HashingTF and CountVectorizer”,
https://mingchen0919.github.io/learning-apache-spark/tf-idf.html (Jun 2017)

[Demers15]
J. DeMers, “Longer Search Queries Are Becoming the Norm: What It Means for SEO”,
https://searchenginewatch.com/sew/opinion/2411478/longer-search-queries-are-becoming-the-norm-what-it-means-for-seo (Jun 2015)

[Egnor14]
Egnor D., R. Lord, “Structured Information Retrieval using XML,”
https://www.research.ibm.com/haifa/sigir00XML/finalpapers/Egnor/ (Nov 2014).

[Elastic14]
“What is elasticsearch,”
http://www.elasticsearch.org/overview/elasticsearch, (Dec 2014).

[Francis09]
Francis S. “Six Logical Writing Structures,”
http://www.writersdigest.com/tip-of-the-day/six-logical-writing-structures (Mar 2009)

[Ho13]
Ho R., “Pragmatic Programming Techniques,”
http://horicky.blogspot.com/2013/02/text-processing-part-2-inverted-index.html,
(Dec 2014).

[Kappa14]
“Fleiss’ kappa,”
http://en.wikipedia.org/wiki/Fleiss%27_kappa (Dec 2014)

[Krawczyk14]
Krawczyk K., “Google is easily the most popular search engine,…”
http://www.digitaltrends.com/web/google-baidu-are-the-worlds-most-popular-search-engines/#ixzz3JoREUjd8, (Jul 2014).

[Math17]
Image of Cosine Similarity [image]. (n.d.). Retrieved from
https://www.mathsisfun.com/algebra/vectors.html (Aug 2017)

[Merriam14] "google" Merriam-Webster.com. 2014. http://www.merriam-webster.com/dictionary/google, (Nov 2014).

[NLP14] “Evaluation of Ranked Retrieval Results” Stanford University https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-ranked-retrieval-results-1.html

[Primmer13]
Primmer R.. 2013. “Structured Vs. Unstructured Data,” http://www.robertPrimmer14.com/blog/structured-vs-unstructured.html (Nov 2015).

[Rogers17]
Rogers I. 2017. “Page Rank Explained: The Google Pagerank Algorithm and How It Works”
http://www.cs.princeton.edu/~chazelle/courses/BIB/pagerank.htm (Oct 2017)

[Sam14]
Fletcher, S., Islam, Md. 2014. “Measuring Information Quality for Privacy Preserving Data Mining. International Journal of Computer Theory and Engineering.” 7. 21-28. 10.7763/IJCTE.2015.V7.924. (2014)

[Solr14]
“Solr is the popular, blazing-fast, open source enterprise search platform built on Apache Lucene,” http://lucene.apache.org/solr/index.html (Dec 2014).

[Strickland14]
Strickland J., “Why is the Google algorithm so important?,” http://computer.howstuffworks.com/google-algorithm1.htm (Nov 2014).

[Thuma15]
Thuma John,
http://www.forbes.com/sites/teradata/2015/01/16/yourmathisallwrongflippingthe8020ruleforanalytics (Jan 2015)

[Trim14]
Trim C., “Natural Language Understanding of Unstructured Data,” https://www.ibm.com/developerworks/community/blogs/nlp/entry/natural_language_understanding_of_unstructured_data1?lang=en (Mar 2014).

[Woord14]
Woord P.,“Class Notes on Semi Structured Data,” http://www.dcs.bbk.ac.uk/~ptw/teaching/ssd/notes.html (Nov 2014).

[Yenan13]
Yenan N., “What is Structured Data and Why Does it Matter for SEO?,”

http://www.adherecreative.com/blog/bid/153475/What-is-Structured-Data-and-Why-Does-it-Matter-for-SEO (Oct 2013).

[Zhou14]. Zhou, Deng Peng. 2006. “Beef Up Web Search Applications with Lucene”. Retrieved June, 2009 from http://www.ibm.com/developerworks/java/library/wa-lucene2/index.html?ca=drs-. (2006)

Appendix A

USE CASE DOCUMENT

In this exercise, the below is all the information you have to go on. Use the provided tool to enter the words, individually or when near other words (like blue van or dog tattoo), in the search input lines provided. Use as many as you think reasonable to ensure you don't miss possible cases – ie catch things that may relate or be the hit you want before searching. Then assess the results of the top 10 per instructions in your video.

Suspect description examples--

1. Looking for a black male with sleeve tattoos up both arms driving a black or dark Chevrolet Impala

2. Looking for a white male with red hair and a beard, riding a blue Harley Davidson motorcycle

3. Looking for a Hispanic female and a light haired, thin, white male couple committing thefts and/or shoplifting

4. Looking for a white male described as a kkk or white supremacy or neo nazi type named Ryan

MO descriptions=

1. Looking for cases where firearms taken in burglary where the perpetrator ransacked the location after cutting or disabling security systems and/or surveillance cameras but no signs of forced entry

2. Looking for cases where auto burglaries and the victim has cash they just took out from the bank - and which they left in the car to run into the store subsequently after leaving the bank - stolen. Also called Jugging.

3. Looking for burglaries where suspects used forcible entry through the roof of businesses to gain access.

4. Looking for apartment burglaries where firearms taken via window entry

Research queries--

1. Doing research on abandoned homes/buildings where drugs and/or homeless are noted

2. Doing research on construction site thefts or burglaries where builders/contractors were unable to provide serial numbers for stolen items or equipment.

Appendix B

USER ANALYSIS DATA EXAMPLE

```
Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Term: abandoned Boost: 1
Term: homes Boost: 1
Term: buildings Boost: 1
Term: drugs Boost: 1
Term: homeless Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201701-3544913914 : 5.776843
Higher : 201701-3551358691 : 5.263109
Much Higher : 201702-3561086240 : 4.995574
Lower : 201701-3544913908 : 4.953643
Lower : 201702-3574765567 : 4.951312
Just Right : 201701-3540800671 : 4.88604
Much Lower : 201702-3574766118 : 4.806753
Much Higher : 201702-3584307275 : 4.718874
Higher : 201702-3581567502 : 4.631528
Higher : 201701-3558931270 : 4.570276
```

Appendix C

PRECISION AND RECALL OF COMPLETE DATA

Mean Average Precision of Results												
	Ranks											
Analyst &Use Case	1	2	3	4	5	6	7	8	9	10	PR Totals	MAP
DL	SD4											
A no boost												
Recall	0	0	0	1	1	1	1	1	1	1	.0625	0.25
Precision	0	0	0	0.25	0.2	0.17	0.14	0.13	0.11	0.1	1 relevant	
B no boost												
Recall	0	0.25	0.5	0.5	0.5	0.75	0.75	0.75	1	1	.373	0.485
Precision	0	0.5	0.33	0.5	0.4	0.5	0.43	0.38	0.44	0.4	4 relevant	
A boosted												
Recall	0.17	0.33	0.33	0.5	0.5	0.5	0.67	0.83	0.83	1	.645	0.703
Precision	1	0.5	0.67	0.75	0.6	0.5	0.57	0.625	0.56	0.6	6 relevant	
B boosted												
Recall	0	0.14	0.29	0.43	0.57	0.57	0.71	0.85	1	1	.59	0.71
Precision	0	0.5	0.67	0.75	0.8	0.67	0.714	0.75	0.78	0.7	7 relevant	
DL	SD1											
A no boost												
Recall	0	0	0	0.33	0.33	0.33	0.67	0.67	0.67	1	.34	0.28
Precision	0	0	0	0.25	0.2	0.17	0.29	0.25	0.22	0.3	3 relevant	
B no boost												
Recall	0	0	0	0	0	0.33	0.67	0.67	1	1	.25	0.263
Precision	0	0	0	0	0	0.17	0.29	0.25	0.33	0.3	3 relevant	
A boosted												
Recall	0	0	0	0	0.33	0.33	0.33	0.67	0.67	1	.2	0.25
Precision	0	0	0	0	0.2	0.17	0.14	0.25	0.22	0.3	3 relevant	
B boosted												
Recall	0	0.11	0.22	0.33	0.44	0.55	0.67	0.78	0.89	1	.56	0.89
Precision	0	0.5	0.67	0.75	0.8	0.83	0.86	0.88	0.89	0.9	9 relevant	

DL	MO3											
A no boost												
Recall	0.25	0.5	0.5	0.5	0.75	0.75	0.75	0.75	0.75	1	.84	0.75
Precision	1	1	0.67	0.5	0.6	0.5	0.43	0.375	0.33	0.4	4 relevant	
B no boost												
Recall	0.13	0.25	0.25	0.38	0.5	0.63	0.75	0.875	1	1	.47	0.875
Precision	1	1	0.67	0.75	0.8	0.83	0.86	0.875	0.89	0.8	8 relevant	
A boosted												
Recall	0.1	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3	0.3	.83	0.83
Precision	1	1	0.67	0.5	0.4	0.5	0.43	0.38	0.33	0.3	3 relevant	
B boosted												
Recall	0	0	0	0	0.1	0.2	0.3	0.3	0.3	0.4	.49	0.34
Precision	0	0	0	0	0.2	0.33	0.43	0.375	0.33	0.4	4 relevant	

MS	RQ1											
A no boost												
Recall	0	0.2	0.2	0.2	0.2	0.4	0.4	0.6	0.8	1	.429	0.429
Precision	0	0.5	0.33	0.25	0.2	0.33	0.29	0.375	0.44	0.5	5 relevant	
B no boost												
Recall	0.14	0.29	0.43	0.57	0.57	0.71	0.86	0.86	1	1	.92	0.92
Precision	1	1	1	1	0.8	0.83	0.86	0.75	0.78	0.7	7 relevant	
A boosted												
Recall	0	0	0	0	0.5	0.5	0.5	1	1	1	.225	0.225
Precision	0	0	0	0	0.2	0.17	0.14	0.25	0.22	0.2	2 relevant	
B boosted												
Recall	0.14	0.29	0.43	0.57	0.57	0.71	0.86	0.86	1	1	.92	0.92
Precision	1	1	1	1	0.8	0.83	0.86	0.75	0.78	0.7	7 relevant	

MS	SD2											
A no boost												
Recall	0.25	0.5	0.75	1	1	1	1	1	1	1	1	1
Precision	1	1	1	1	0.8	0.67	0.57	0.5	0.44	0.4	4 relevant	
B no boost												
Recall	0	0.11	0.22	0.33	0.44	0.56	0.67	0.78	0.89	1	.786	0.786
Precision	0	0.5	0.67	0.75	0.8	0.83	0.86	0.875	0.89	0.9	9 relevant	

A boosted												
Recall	0	0	0.17	0.33	0.5	0.67	0.833	1	1	1	.593	0.593
Precision	0	0	0.33	0.5	0.6	0.67	0.71	0.75	0.67	0.6	6 relevant	
B boosted												
Recall	0	0.11	0.22	0.33	0.44	0.56	0.67	0.78	0.89	1	.786	0.786
Precision	0	0.5	0.67	0.75	0.8	0.83	0.86	0.875	0.89	0.9	9 relevant	
MS	SD3											
A no boost												
Recall	0	0	0	0.25	0.25	0.5	0.5	0.75	0.75	1	.338	0.338
Precision	0	0	0	0.25	0.2	0.33	0.29	0.375	0.33	0.4	4 relevant	
B no boost												
Recall	0	0	0	0	0.33	0.33	0.33	0.67	1	1	.373	0.373
Precision	0	0	0	0	0.2	0.17	0.14	0.25	0.67	0.3	3 relevant	
A boosted												
Recall	0	0	0	0	0	0	0	0.33	0.67	1	.215	0.215
Precision	0	0	0	0	0	0	0	0.125	0.22	0.3	3 relevant	
B boosted												
Recall	0	0	0	0	0.33	0.33	0.33	67	1	1	.26	0.26
Precision	0	0	0	0	0.2	0.17	0.14	0.25	0.33	0.3	3 relevant	
MS	MO2											
A no boost												
Recall	0	0.17	0.17	0.17	0.17	0.33	0.5	0.67	0.83	1	.486	0.486
Precision	0	0.5	0.33	0.25	0.2	0.33	0.43	0.5	0.56	0.6	6 relevant	
B no boost												
Recall	0.25	0.25	0.25	0.5	0.75	0.75	0.75	1	1	1	.65	0.65
Precision	1	0.5	0.33	0.5	0.6	0.5	0.43	0.5	0.44	0.4	4 relevant	
A boosted												
Recall	0	0	0.2	0.4	0.4	0.4	0.6	0.83	1	1	.464	0.464
Precision	0	0	0.33	0.5	0.4	0.33	0.43	0.5	0.56	0.5	5 relevant	
B boosted												
Recall	0.13	0.25	0.38	0.5	0.63	0.75	0.75	0.875	1	1	.97	0.97
Precision	1	1	1	1	1	1	0.86	0.875	0.89	0.8	8 relevant	

Overall MAP for scoring methods A and B

MAP

A no boost	50.47%
A boosted	46.85%
B no boost	62.17%
B boosted	69.66%

Appendix D

RAW USER ANALYSIS DATA

```
Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Phrase: auto burglaries Boost: 1
Phrase: vehicle burglary Boost: 1
Phrase: grand theft Boost: 1
Phrase: grand theft auto Boost: 1
Term: cash Boost: 1
Term: bank Boost: 1
Term: jugging Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201701-3540847846 : 35.99052
Lower : 201701-3550960324 : 27.8518
Lower : 201701-3551029348 : 26.36259
Lower : 201701-3551115864 : 19.02556
Just Right : 201701-3551074451 : 16.93837
Lower : 201701-3580958019 : 15.91531
Lower : 201701-3534151981 : 15.5343
Lower : 201701-3552994379 : 15.37818
Lower : 201702-3568979160 : 15.26949
Lower : 201701-3534344233 : 15.22045
 : 201701-3550195408 : 14.11531
 : 201701-3543703391 : 13.85799
 : 201701-3550106490 : 13.76833
 : 201701-3531502434 : 13.31789
 : 201701-3547437907 : 13.27941
 : 201702-3575751677 : 13.25168
 : 201702-3581734827 : 12.81135
 : 201701-3533222307 : 12.42744
 : 201701-3550197705 : 12.32712
 : 201701-3538562158 : 11.8573
 : 201702-3570047860 : 11.53363
 : 201701-3543653944 : 11.47361
 : 201701-3531548675 : 11.41534
 : 201701-3547437911 : 11.40378
 : 201701-3551933275 : 11.25383
 : 201701-3540295058 : 11.22918
 : 201702-3570115084 : 11.14923
 : 201701-3531052688 : 10.86262
 : 201702-3568109984 : 10.79716
 : 201701-3534450639 : 10.68593

Score Type: Analysis Score Form A Analysis Number 1
```

```
Query Detail:
Term: burglaries Boost: 1
Phrase: break in Boost: 1
Phrase: forcible entry Boost: 1
Phrase: roof entry Boost: 1
Phrase: business burglary Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201701-3581648849 : 11.74756
Just Right : 201701-3560000309 : 10.6589
Much Lower : 201702-3564904796 : 7.334181
Much Lower : 201701-3544602378 : 6.022136
Much Higher : 201702-3570049703 : 5.833292
Lower : 201701-3551292812 : 5.806771
Lower : 201702-3573648162 : 5.386363
Lower : 201701-3560000261 : 5.133927
Lower : 201701-3577404482 : 5.118783
Just Right : 201701-3529605662 : 4.645417
 : 201701-3552562608 : 4.491251
 : 201701-3533649312 : 4.400509
 : 201701-3546441409 : 4.258293
 : 201701-3530852854 : 4.148839
 : 201701-3528269043 : 4.06474
 : 201701-3544490622 : 4.03406
 : 201701-3532224613 : 4.023049
 : 201702-3581152157 : 3.972468
 : 201701-3544491355 : 3.766607
 : 201701-3547438419 : 3.699007
 : 201702-3577404646 : 3.683458
 : 201701-3553047388 : 3.627927
 : 201702-3570048625 : 3.62635
 : 201701-3531671388 : 3.484063
 : 201701-3530935200 : 3.406635
 : 201701-3546441151 : 3.309745
 : 201701-3540418209 : 3.243506
 : 201701-3539329794 : 3.017287
 : 201701-3548066840 : 3.011068
 : 201701-3558701870 : 2.933672

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Phrase: construction site thefts Boost: 1
Phrase: construction burglaries Boost: 1
Phrase: stolen construction equipment Boost: 1
Phrase: new build neighborhoods Boost: 1
Phrase: construction thefts Boost: 1

The number of records found: 9
The index used: C:\\indexes
The number of documents in the index is: 1641816
```

```
Just Right : 201701-3572268274 : 24.43997
Just Right : 201701-3570432463 : 8.533347
Lower : 201701-3566949683 : 7.542485
Much Lower : 201701-3576983744 : 7.466679
Much Lower : 201701-3582343657 : 7.390096
Much Lower : 201701-3562562618 : 5.66007
Higher : 201702-3580316330 : 5.213573
Lower : 201701-3560985485 : 2.474575
Lower : 201701-3547314543 : 2.449704

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Phrase: black male Boost: 1
Phrase: african american male Boost: 1
Phrase: sleeve tattoos Boost: 1
Phrase: arm tattoos Boost: 1
Phrase: black chevrolet impala Boost: 1
Phrase: dark chevrolet impala Boost: 1
Phrase: chevy impala Boost: 1
Phrase: black impala Boost: 1
Phrase: chevy sedan Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201701-3562867902 : 26.26751
Lower : 201701-3551971669 : 18.42052
Much Lower : 201701-3578159854 : 17.69795
Much Higher : 201702-3574755931 : 16.6746
Much Lower : 201701-3531613176 : 14.48115
Lower : 201701-3558928891 : 14.21966
Higher : 201702-3568028310 : 14.08465
Much Lower : 201701-3553824229 : 13.93236
Much Lower : 201702-3565649617 : 13.65296
Just Right : 201702-3568028249 : 12.86341
 : 201702-3584530034 : 12.575
 : 201701-3552202232 : 12.51593
 : 201702-3566461009 : 12.28284
 : 201701-3548165189 : 12.06762
 : 201701-3533064191 : 11.94634
 : 201701-3530084891 : 11.85582
 : 201702-3580782744 : 11.59806
 : 201701-3554540930 : 11.37572
 : 201701-3548054371 : 11.35681
 : 201702-3581881996 : 10.34583
 : 201702-3581637338 : 10.24009
 : 201702-3571380314 : 9.71638
 : 201701-3532464912 : 9.555425
 : 201701-3575610634 : 9.044511
 : 201701-3539854802 : 8.902709
 : 201701-3569977157 : 8.832048
 : 201701-3569215131 : 8.588672
 : 201702-3572414767 : 8.445882
 : 201701-3531613404 : 8.400619
 : 201702-3572143783 : 8.383334

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Phrase: white male Boost: 1
Phrase: caucasion male Boost: 1
Term: wm Boost: 1
Term: kkk Boost: 1
Phrase: white supremacy Boost: 1
Phrase: neo nazi Boost: 1
Term: ryan Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Much Lower : 201701-3581571429 : 19.37165
Much Lower : 201701-3553645402 : 18.64038
Much Lower : 201701-3555036327 : 14.9123
Higher : 201701-3533533466 : 12.78517
Much Lower : 201701-3569969826 : 11.18423
Much Lower : 201702-3569756560 : 8.843594
Much Lower : 201701-3538060958 : 7.180372
Lower : 201701-3531613188 : 6.321493
Much Lower : 201701-3531409708 : 6.132749
Much Lower : 201702-3573706320 : 6.092748
 : 201701-3558705361 : 5.661473
 : 201701-3581571421 : 5.603406
 : 201701-3530610680 : 5.592705
 : 201702-3570190760 : 5.536495
 : 201701-3531469096 : 5.504737
 : 201701-3534083686 : 5.420635
 : 201702-3561091676 : 5.393464
 : 201701-3581736460 : 5.125072
 : 201702-3580322447 : 5.077778
 : 201702-3561091607 : 4.862317
 : 201701-3530908912 : 4.848364
 : 201701-3529297839 : 4.745567
 : 201702-3581882064 : 4.563486
 : 201701-3543015308 : 4.470055
 : 201702-3583429511 : 4.443185
 : 201701-3530380078 : 4.426643
 : 201702-3583537911 : 4.411004
 : 201701-3551971645 : 4.347083
 : 201701-3530935310 : 4.285388
 : 201701-3538496722 : 4.268725

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Phrase: auto burglaries Boost: 1
Phrase: vehicle burglary Boost: 1
Phrase: grand theft Boost: 1
```

```
Phrase: grand theft auto Boost: 1
Term: cash Boost: 1
Term: bank Boost: 1
Term: jugging Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201701-3580020809 : 5.02862
Just Right : 201701-3571201305 : 5.025957
Lower : 201702-3582374538 : 5.016291
Lower : 201702-3581567347 : 5.014108
Just Right : 201701-3537320893 : 4.023062
Just Right : 201701-3547437911 : 4.022977
Just Right : 201701-3544915216 : 4.022325
Lower : 201702-3580782585 : 4.015129
Higher : 201702-3582769373 : 4.014818
Higher : 201702-3571613414 : 4.012662
 : 201702-3574906360 : 4.012537
 : 201702-3579681363 : 4.012472
 : 201702-3570115318 : 4.011907
 : 201701-3566568263 : 4.008163
 : 201701-3543703391 : 3.033819
 : 201701-3551074451 : 3.026381
 : 201702-3579639343 : 3.026253
 : 201702-3581881822 : 3.025064
 : 201701-3558922401 : 3.023364
 : 201701-3532353801 : 3.02267
 : 201702-3581734827 : 3.0226
 : 201701-3559801437 : 3.022217
 : 201701-3550195408 : 3.021984
 : 201702-3569971943 : 3.021697
 : 201701-3538562158 : 3.021166
 : 201702-3565603181 : 3.020042
 : 201701-3560773628 : 3.019794
 : 201701-3565645963 : 3.019307
 : 201701-3548054504 : 3.019167
 : 201701-3554541020 : 3.018806

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Term: burglaries Boost: 1
Phrase: break in Boost: 1
Phrase: forcible entry Boost: 1
Phrase: roof entry Boost: 1
Phrase: business burglary Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201702-3577404646 : 3.001471
Just Right : 201701-3563752671 : 3.00114
Lower : 201701-3564994855 : 3.001074
```

```
Higher : 201701-3547438419 : 3.00095
Much Higher : 201701-3581648849 : 2.044127
Much Higher : 201701-3560000309 : 2.040767
Higher : 201702-3570049703 : 2.026017
Higher : 201701-3546441151 : 2.019741
Just Right : 201701-3544491355 : 2.019344
Lower : 201701-3532530493 : 2.016504
 : 201701-3562752768 : 2.012234
 : 201701-3553047388 : 2.011011
 : 201702-3573648160 : 2.010518
 : 201701-3543192605 : 2.010377
 : 201702-3571470696 : 2.010309
 : 201702-3583113277 : 2.009725
 : 201701-3530212289 : 2.009389
 : 201701-3581648840 : 2.008817
 : 201702-3574765504 : 2.008156
 : 201702-3579681194 : 2.007386
 : 201702-3582641426 : 2.006331
 : 201701-3563752689 : 2.005142
 : 201701-3563752029 : 2.003673
 : 201701-3543221430 : 2.001899
 : 201702-3579617007 : 2.001551
 : 201701-3551971513 : 2.001396
 : 201701-3538294014 : 2.001343
 : 201701-3569359227 : 2.001329
 : 201702-3582060576 : 2.001163
 : 201702-3580078297 : 2.001073

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Phrase: construction site thefts Boost: 1
Phrase: construction burglaries Boost: 1
Phrase: stolen construction equipment Boost: 1
Phrase: new build neighborhoods Boost: 1
Phrase: construction thefts Boost: 1

The number of records found: 9
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201701-3572268274 : 3.015895
Lower : 201701-3570432463 : 2.011639
Lower : 201701-3566949683 : 2.010287
Lower : 201701-3576983744 : 2.010184
Lower : 201701-3582343657 : 2.010079
Much Higher : 201702-3580316330 : 1.009519
Much Lower : 201701-3562562618 : 1.003906
Lower : 201701-3560985485 : 1.001708
Lower : 201701-3547314543 : 1.001691

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Phrase: black male Boost: 1
```

```
Phrase: african american male Boost: 1
Phrase: sleeve tattoos Boost: 1
Phrase: arm tattoos Boost: 1
Phrase: black chevrolet impala Boost: 1
Phrase: dark chevrolet impala Boost: 1
Phrase: chevy impala Boost: 1
Phrase: black impala Boost: 1
Phrase: chevy sedan Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201702-3568028310 : 4.032598
Lower : 201702-3580782744 : 4.026371
Lower : 201702-3568028249 : 4.000277
Much Lower : 201701-3562867902 : 3.03498
Much Lower : 201702-3583429481 : 3.014097
Higher : 201702-3566461009 : 3.012762
Just Right : 201702-3566456690 : 3.011824
Lower : 201701-3578159854 : 3.011385
Just Right : 201701-3546441267 : 3.008775
Much Lower : 201701-3577404468 : 3.008683
 : 201701-3575442527 : 3.008273
 : 201702-3574755931 : 3.000477
 : 201701-3551971669 : 3.000377
 : 201702-3572143546 : 3.000262
 : 201701-3530084891 : 3.00022
 : 201702-3571380314 : 3.000193
 : 201702-3572143783 : 3.000167
 : 201702-3569359435 : 3.000167
 : 201701-3531613176 : 2.016959
 : 201702-3565649617 : 2.015989
 : 201701-3533064191 : 2.013991
 : 201702-3581881996 : 2.012116
 : 201701-3569977157 : 2.01126
 : 201701-3569215131 : 2.010949
 : 201701-3554369484 : 2.010629
 : 201702-3578270001 : 2.01022
 : 201701-3581881949 : 2.010041
 : 201701-3575579518 : 2.009662
 : 201702-3582480497 : 2.008432
 : 201701-3554107597 : 2.008126

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Phrase: white male Boost: 1
Phrase: caucasion male Boost: 1
Term: wm Boost: 1
Term: kkk Boost: 1
Phrase: white supremacy Boost: 1
Phrase: neo nazi Boost: 1
Term: ryan Boost: 1
```

```
The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Much Lower : 201701-3565732115 : 3.001185
Higher : 201701-3533533466 : 2.032932
Lower : 201701-3530380078 : 2.011402
Just Right : 201702-3583537911 : 2.011362
Lower : 201701-3538496722 : 2.010995
Just Right : 201701-3555053602 : 2.010429
Lower : 201701-3533528729 : 2.010351
Lower : 201702-3578385967 : 2.010069
Just Right : 201701-3530852856 : 2.01
Lower : 201702-3581997465 : 2.009522
 : 201701-3555036184 : 2.009218
 : 201701-3558702085 : 2.008667
 : 201702-3582641547 : 2.008589
 : 201701-3558684975 : 2.008246
 : 201702-3581567726 : 2.008243
 : 201701-3530611179 : 2.007775
 : 201701-3553127386 : 2.007645
 : 201701-3553971032 : 2.00759
 : 201701-3542747825 : 2.007267
 : 201701-3543015308 : 2.007171
 : 201701-3538063031 : 2.007158
 : 201701-3530629543 : 2.007143
 : 201701-3552304792 : 2.007142
 : 201702-3570433440 : 2.007054
 : 201701-3528925183 : 2.006872
 : 201701-3551688029 : 2.00675
 : 201702-3579734357 : 2.00674
 : 201701-3555037660 : 2.006713
 : 201701-3561203382 : 2.006599
 : 201701-3528995178 : 2.00656

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Phrase: auto burglary Boost: 1
Term: bank Boost: 1
Term: store Boost: 1
Term: jugging Boost: 1
Term: car Boost: 1
Term: cash Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201701-3547314974 : 7.030466
Lower : 201702-3568398555 : 6.9853
Lower : 201702-3581648890 : 6.880278
Lower : 201702-3581567514 : 6.526258
Lower : 201701-3563326643 : 6.41352
Lower : 201702-3579454033 : 6.333113
Higher : 201702-3572414819 : 6.11609
```

```
Lower : 201702-3582769291 : 6.108038
Higher : 201702-3579454036 : 6.03847
Lower : 201701-3552295848 : 5.95122
 : 201702-3574765362 : 5.936357
 : 201702-3569876106 : 5.853105
 : 201702-3574764885 : 5.844195
 : 201702-3579639343 : 5.710808
 : 201702-3582060580 : 5.545459
 : 201701-3553159764 : 5.505538
 : 201702-3581881822 : 5.452127
 : 201701-3533064306 : 5.431349
 : 201702-3581298354 : 5.323083
 : 201701-3553047313 : 5.307956
 : 201702-3572895997 : 5.29318
 : 201702-3572143810 : 5.134067
 : 201702-3583178298 : 5.131498
 : 201702-3564994922 : 5.12922
 : 201701-3554637114 : 5.062512
 : 201701-3531389794 : 5.045067
 : 201702-3579454034 : 4.996798
 : 201701-3538562491 : 4.993088
 : 201701-3555037090 : 4.949689
 : 201701-3564994812 : 4.943706

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Phrase: window entry Boost: 1
Term: window Boost: 1
Phrase: apartment burglary Boost: 1
Term: apartment Boost: 1
Term: burglary Boost: 1
Term: firearms Boost: 1
Term: handgun Boost: 1
Term: rifle Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201701-3548066745 : 9.716422
Much Higher : 201701-3575579563 : 9.629402
Lower : 201701-3532451863 : 9.097729
Lower : 201702-3578259538 : 8.874594
Lower : 201701-3538293937 : 8.808522
Just Right : 201702-3577404662 : 8.686251
Lower : 201701-3562752503 : 8.52562
Just Right : 201702-3580875137 : 8.187551
Higher : 201701-3531683316 : 8.048148
Higher : 201701-3574765765 : 7.993158
 : 201701-3532353849 : 7.945142
 : 201701-3537607129 : 7.886965
 : 201702-3578151784 : 7.844804
 : 201701-3568272726 : 7.684978
 : 201702-3564995192 : 7.62826
 : 201702-3572896134 : 7.49424
 : 201702-3566461041 : 7.395838
 : 201702-3571877088 : 7.326088
 : 201701-3529405785 : 7.270479
 : 201702-3572259758 : 7.257072
 : 201702-3575573470 : 7.218037
 : 201701-3537527910 : 7.180001
 : 201701-3531895240 : 7.162064
 : 201701-3552840485 : 7.160283
 : 201701-3543032024 : 7.143662
 : 201702-3560789862 : 7.13447
 : 201702-3582480561 : 7.099667
 : 201701-3569876405 : 7.065568
 : 201701-3553047503 : 7.034581
 : 201702-3574754867 : 7.014654

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Term: abandoned Boost: 1
Phrase: abandoned building Boost: 1
Phrase: abandoned house Boost: 1
Phrase: abandonded home Boost: 1
Term: homeless Boost: 1
Term: drugs Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201701-3551358691 : 14.78805
Lower : 201701-3546676978 : 9.542046
Just Right : 201701-3538293701 : 8.823707
Lower : 201701-3530146007 : 8.809452
Just Right : 201701-3551971608 : 8.747096
Lower : 201701-3547516645 : 8.691321
Much Higher : 201701-3543819748 : 8.654785
Much Higher : 201702-3583178342 : 8.440416
Lower : 201701-3530216328 : 7.791047
Much Higher : 201701-3530092155 : 7.649822
 : 201701-3552562720 : 7.526905
 : 201702-3579685340 : 7.406905
 : 201702-3564904968 : 6.817167
 : 201701-3543846538 : 6.720707
 : 201701-3529411666 : 6.42687
 : 201701-3538277758 : 6.358472
 : 201701-3547314687 : 6.070491
 : 201701-3532358439 : 5.933519
 : 201702-3571270484 : 5.880619
 : 201701-3552049622 : 5.843286
 : 201702-3580514113 : 5.811028
 : 201701-3533064214 : 5.628753
 : 201701-3543837517 : 5.530823
 : 201702-3575442874 : 5.447434
 : 201701-3584528594 : 5.04053
 : 201701-3560085651 : 4.983378
 : 201702-3569977206 : 4.922612
 : 201701-3540800671 : 4.88604
 : 201701-3558931270 : 4.570276
 : 201702-3582060042 : 4.549453

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Phrase: white male Boost: 1
Term: wm Boost: 1
Phrase: red hair Boost: 1
Term: beard Boost: 1
Phrase: blue harley davidson Boost: 1
Phrase: blue harley Boost: 1
Phrase: harley davidson Boost: 1
Term: harley Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201701-3530371215 : 39.90197
Lower : 201701-3555053432 : 27.47263
Higher : 201701-3530845632 : 19.53661
Lower : 201701-3552049549 : 15.71445
Lower : 201701-3547594465 : 13.46952
Lower : 201702-3581649068 : 11.66495
Lower : 201701-3532451873 : 11.59272
Lower : 201701-3537625575 : 11.2246
Lower : 201701-3552082132 : 9.720793
Higher : 201701-3529411634 : 9.164851
 : 201701-3534463022 : 8.979683
 : 201701-3553153934 : 8.523228
 : 201701-3553624971 : 8.397013
 : 201702-3581881923 : 7.936994
 : 201701-3581648853 : 7.272026
 : 201701-3538060958 : 7.180372
 : 201701-3553824104 : 6.71761
 : 201701-3531613188 : 6.321493
 : 201701-3561085606 : 6.217722
 : 201701-3531409708 : 6.132749
 : 201702-3573706320 : 6.092748
 : 201701-3559022722 : 6.016068
 : 201702-3569752026 : 5.731232
 : 201701-3544913582 : 5.721392
 : 201701-3530610680 : 5.592705
 : 201701-3553824274 : 5.554546
 : 201702-3570190760 : 5.536495
 : 201701-3531469096 : 5.504737
 : 201701-3534083686 : 5.420635
 : 201702-3561091676 : 5.393464

Score Type: Analysis Score Form A Analysis Number 1
```

```
Query Detail:
Phrase: hispanic female Boost: 1
Phrase: white male Boost: 1
Term: hf Boost: 1
Term: wm Boost: 1
Term: thin Boost: 1
Term: shoplifting Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201701-3538060958 : 8.928061
Lower : 201702-3581882033 : 7.673116
Lower : 201702-3578270012 : 6.503708
Lower : 201701-3560296370 : 6.45325
Lower : 201701-3531613188 : 6.321493
Lower : 201701-3547070574 : 6.20891
Lower : 201701-3531409708 : 6.132749
Lower : 201702-3573706320 : 6.092748
Higher : 201701-3552088795 : 5.970214
Lower : 201702-3569692747 : 5.735067
 : 201702-3573648143 : 5.664509
 : 201701-3532466453 : 5.613746
 : 201701-3530610680 : 5.592705
 : 201702-3583113136 : 5.585819
 : 201702-3568045033 : 5.552446
 : 201702-3570190760 : 5.536495
 : 201701-3531469096 : 5.504737
 : 201701-3534083686 : 5.420635
 : 201701-3574532801 : 5.410642
 : 201702-3561091676 : 5.393464
 : 201701-3533272388 : 5.255116
 : 201701-3529469181 : 5.121023
 : 201701-3537818501 : 5.119236
 : 201702-3580322447 : 5.077778
 : 201701-3555230079 : 5.070525
 : 201701-3552562924 : 5.017558
 : 201702-3573182182 : 4.973493
 : 201701-3530908912 : 4.848364
 : 201701-3560660723 : 4.782377
 : 201701-3529297839 : 4.745567

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Term: roof Boost: 1
Term: business Boost: 2
Term: burglary Boost: 1

The number of records found: 10
The index used: C:\\indexes
The number of documents in the index is: 1641816
Much Higher : 201702-3570432781 : 6.135654
Much Higher : 201701-3540821223 : 5.079417
Much Higher : 201702-3581089767 : 4.908524
Much Higher : 201701-3546308694 : 4.321653
Much Lower : 201701-3581648849 : 4.115605
Much Higher : 201702-3582071160 : 3.981427
Much Lower : 201702-3568272913 : 3.953999
Much Higher : 201701-3529469197 : 3.90245
Much Lower : 201701-3560000309 : 3.673982
Lower : 201701-3553047388 : 3.636667

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Term: apartment Boost: 1
Term: apt Boost: 1
Term: complex Boost: 1
Term: firearms Boost: 1
Term: burglary Boost: 1

The number of records found: 10
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201701-3537528202 : 6.599075
Much Higher : 201701-3538277815 : 5.919493
Just Right : 201701-3548066745 : 5.768955
Just Right : 201701-3551971493 : 5.41482
Higher : 201701-3553047503 : 5.364847
Much Higher : 201701-3531883026 : 5.359215
Much Higher : 201701-3532353849 : 5.352892
Much Higher : 201701-3538277466 : 5.346678
Much Higher : 201701-3533809616 : 5.319809
Much Higher : 201701-3531751456 : 5.309621

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Term: abandoned Boost: 1
Term: drugs Boost: 1
Term: homeless Boost: 1

The number of records found: 10
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201701-3551358691 : 5.263109
Much Higher : 201701-3540800671 : 4.88604
Just Right : 201701-3558931270 : 4.570276
Much Higher : 201702-3582060042 : 4.549453
Much Higher : 201702-3578333605 : 4.522359
Just Right : 201702-3583178342 : 4.230949
Much Higher : 201702-3574766182 : 3.939941
Just Right : 201701-3538063287 : 3.785686
Much Higher : 201701-3539536396 : 3.762661
Higher : 201702-3577567442 : 3.530607

Score Type: Analysis Score Form A Analysis Number 1
```

```
Query Detail:
Phrase: black male Boost: 2
Term: bm Boost: 2
Term: impala Boost: 1
Term: tattoo Boost: 1

The number of records found: 10
The index used: C:\\indexes
The number of documents in the index is: 1641816
Much Higher : 201701-3575610634 : 18.08902
Much Higher : 201701-3554370121 : 14.6046
Much Higher : 201702-3581648950 : 12.79087
Much Higher : 201701-3543846384 : 11.67641
Lower : 201702-3574755931 : 11.31612
Much Higher : 201701-3534196144 : 10.80235
Much Higher : 201701-3539348024 : 10.72635
Much Higher : 201701-3532881633 : 10.65906
Much Higher : 201702-3568933846 : 10.5591
Much Higher : 201701-3530092155 : 10.29763

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Phrase: white male Boost: 1
Term: wm Boost: 1
Term: kkk Boost: 1
Term: supremacy Boost: 1
Term: nazi Boost: 1
Term: ryan Boost: 1

The number of records found: 10
The index used: C:\\indexes
The number of documents in the index is: 1641816
Much Higher : 201702-3569756560 : 8.843594
Much Lower : 201701-3533533466 : 8.607029
Much Higher : 201701-3538060958 : 7.180372
Much Higher : 201701-3558705361 : 6.634763
Much Higher : 201701-3581571421 : 6.566712
Much Higher : 201701-3531613188 : 6.321493
Much Higher : 201701-3531409708 : 6.132749
Much Higher : 201702-3573706320 : 6.092748
Much Higher : 201701-3530610680 : 5.592705
Much Higher : 201702-3570190760 : 5.536495

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Term: roof Boost: 1
Term: business Boost: 2
Term: burglary Boost: 1

The number of records found: 10
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201701-3581648849 : 3.153404
Just Right : 201702-3568272913 : 3.147381
Much Higher : 201701-3529469197 : 3.145459
Just Right : 201701-3560000309 : 3.136943
Just Right : 201701-3553047388 : 3.135552
Just Right : 201701-3531406232 : 3.126994
Much Higher : 201702-3561085513 : 3.124151
Just Right : 201701-3563752671 : 3.121326
Just Right : 201702-3571379749 : 3.120511
Higher : 201702-3582480366 : 3.119299

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Term: apartment Boost: 1
Term: apt Boost: 1
Term: complex Boost: 1
Term: firearms Boost: 1
Term: burglary Boost: 1

The number of records found: 10
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201701-3562752533 : 5.07508
Just Right : 201701-3580322333 : 5.048823
Just Right : 201701-3578269970 : 5.048571
Just Right : 201701-3559044198 : 5.048314
Just Right : 201701-3553916947 : 5.048051
Just Right : 201701-3537528202 : 4.083709
Just Right : 201701-3548066745 : 4.073179
Higher : 201701-3553047503 : 4.068053
Much Higher : 201701-3531883026 : 4.067982
Higher : 201701-3532603252 : 4.067051

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Term: abandoned Boost: 1
Term: drugs Boost: 1
Term: homeless Boost: 1

The number of records found: 10
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201702-3583178342 : 3.143297
Just Right : 201702-3562745461 : 3.10252
Just Right : 201701-3551358691 : 2.118836
Just Right : 201701-3558931270 : 2.103193
Much Higher : 201702-3582060042 : 2.102723
Much Higher : 201702-3574766182 : 2.08896
Just Right : 201701-3538063287 : 2.085478
Much Higher : 201701-3539536396 : 2.084958
Much Higher : 201702-3577567442 : 2.079718
Much Higher : 201702-3562877936 : 2.074107

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Phrase: white male Boost: 1
Term: wm Boost: 1
Term: kkk Boost: 1
Term: supremacy Boost: 1
Term: nazi Boost: 1
Term: ryan Boost: 1

The number of records found: 10
The index used: C:\\indexes
The number of documents in the index is: 1641816
Much Higher : 201701-3565732115 : 3.040433
Much Higher : 201701-3538060958 : 2.114327
Much Higher : 201701-3530610680 : 2.088557
Much Higher : 201702-3573706320 : 2.086768
Much Higher : 201702-3563006239 : 2.079208
Much Higher : 201702-3566693832 : 2.078412
Much Higher : 201701-3551757998 : 2.076999
Much Higher : 201701-3529297839 : 2.075143
Much Higher : 201702-3581879693 : 2.074715
Much Higher : 201701-3554547472 : 2.07186

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Phrase: black male Boost: 2
Term: bm Boost: 2
Term: impala Boost: 1
Term: tattoo Boost: 1

The number of records found: 10
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201702-3574765862 : 3.039201
Just Right : 201702-3580078292 : 3.034789
Lower : 201702-3580074366 : 3.033859
Just Right : 201701-3546441154 : 3.030164
Just Right : 201702-3571380522 : 3.029931
Much Higher : 201702-3582480565 : 3.028776
Much Higher : 201701-3562752574 : 3.028053
Much Higher : 201701-3534196144 : 2.082091
Lower : 201702-3574755931 : 2.08197
Much Higher : 201702-3583987323 : 2.069554

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Phrase: auto burglary Boost: 1
Term: cash Boost: 1
Term: atm Boost: 1
```

```
Term: jugging Boost: 1
Term: stolen Boost: 1
Term: car Boost: 1
Term: vehicle Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201702-3581879441 : 7.453504
Just Right : 201701-3553047313 : 7.097847
Lower : 201702-3569969795 : 7.012037
Lower : 201702-3572259798 : 7.012036
Lower : 201702-3568398555 : 6.9853
Higher : 201702-3574765362 : 6.848553
Higher : 201702-3560991446 : 6.662108
Higher : 201701-3552295848 : 6.645565
Just Right : 201701-3569109648 : 6.464915
Higher : 201702-3572895997 : 6.266195
 : 201701-3578159887 : 6.194795
 : 201701-3531883026 : 6.043301
 : 201701-3572896127 : 5.957431
 : 201701-3553159764 : 5.951952
 : 201701-3572143544 : 5.851605
 : 201702-3562745379 : 5.817589
 : 201702-3574765354 : 5.790452
 : 201701-3558937012 : 5.786562
 : 201701-3531613111 : 5.706643
 : 201701-3531052688 : 5.681178
 : 201701-3532603252 : 5.655043
 : 201702-3572414819 : 5.516006
 : 201701-3559179448 : 5.502613
 : 201701-3566460855 : 5.455623
 : 201702-3581648890 : 5.452483
 : 201701-3577404510 : 5.451762
 : 201701-3563326389 : 5.447851
 : 201702-3576105656 : 5.440594
 : 201702-3574765866 : 5.415101
 : 201702-3574765885 : 5.414778

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Term: apartment Boost: 1
Term: burglaries Boost: 1
Term: firearms Boost: 1
Term: stolen Boost: 1
Term: taken Boost: 1
Phrase: window entry Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201701-3532451863 : 6.880041
Lower : 201702-3575442715 : 6.195412
```

```
Just Right : 201701-3531683316 : 6.020036
Higher : 201701-3569876405 : 5.939603
Lower : 201701-3531895240 : 5.702696
Higher : 201702-3572896134 : 5.638065
Much Higher : 201702-3581571737 : 5.535788
Higher : 201701-3546441120 : 5.438508
Just Right : 201701-3551971493 : 5.401882
Much Higher : 201701-3574765765 : 5.38009
 : 201701-3530613251 : 5.358103
 : 201701-3532466348 : 5.341593
 : 201701-3546441117 : 5.298488
 : 201701-3537607129 : 5.291805
 : 201702-3564995192 : 5.284733
 : 201702-3563752049 : 5.272098
 : 201702-3574754867 : 5.196121
 : 201702-3583205172 : 5.183365
 : 201701-3553047503 : 5.161823
 : 201702-3561144959 : 5.138896
 : 201701-3532358273 : 5.118524
 : 201702-3582378258 : 5.113741
 : 201701-3562752503 : 5.095383
 : 201702-3582640933 : 5.070526
 : 201701-3564994815 : 4.989927
 : 201701-3538293937 : 4.939809
 : 201701-3544228481 : 4.927369
 : 201701-3546548164 : 4.903367
 : 201701-3546441124 : 4.891954
 : 201702-3581152157 : 4.869029

Score Type: Analysis Score Form A Analysis Number 2

Query Detail:
Phrase: white male Boost: 1
Phrase: red hair Boost: 1
Phrase: red beard Boost: 1
Phrase: blue harley Boost: 1
Phrase: blue harley davidson Boost: 1
Phrase: blue motorcycle Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Much Higher : 201701-3530371215 : 28.90414
Much Higher : 201701-3530380480 : 11.5082
Much Higher : 201701-3555053432 : 10.52384
Much Higher : 201701-3530845632 : 8.865854
Lower : 201702-3580755972 : 7.021039
Lower : 201702-3575296581 : 6.854345
Higher : 201701-3531613188 : 6.321493
Higher : 201701-3531409708 : 6.132749
Higher : 201701-3534344292 : 6.06849
Just Right : 201701-3537818541 : 6.009704
 : 201701-3528767150 : 5.453973
 : 201701-3534083686 : 5.420635
 : 201702-3561091676 : 5.393464
 : 201701-3547595305 : 5.265779
 : 201701-3561085606 : 5.250021
 : 201701-3537527900 : 5.106303
 : 201701-3559022722 : 5.103711
 : 201702-3580322447 : 5.077778
 : 201701-3530908912 : 4.848364
 : 201701-3553351184 : 4.807763
 : 201701-3553624971 : 4.793535
 : 201702-3581882064 : 4.563486
 : 201701-3582060456 : 4.560299
 : 201702-3583429511 : 4.443185
 : 201701-3551971645 : 4.347083
 : 201701-3530935310 : 4.285388
 : 201701-3530611211 : 4.21542
 : 201701-3532358245 : 4.068764
 : 201701-3543846636 : 4.065476
 : 201702-3578270081 : 3.973796

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Term: hispanic Boost: 1
Term: latina Boost: 1
Term: latin Boost: 1
Term: female Boost: 1
Term: h/f Boost: 1
Term: l/f Boost: 1
Term: light Boost: 1
Term: blonde Boost: 1
Term: male Boost: 1
Term: thin Boost: 1
Term: skinny Boost: 1
Term: committing Boost: 1
Term: thefts Boost: 1
Term: stealing Boost: 1
Term: shoplifitng Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201702-3576647914 : 11.13883
Lower : 201702-3562752881 : 7.607617
Lower : 201701-3530720173 : 7.521863
Higher : 201701-3575442539 : 7.004695
Lower : 201702-3569741579 : 6.871832
Higher : 201701-3560086702 : 6.383531
Lower : 201701-3555136555 : 6.32455
Higher : 201701-3560000304 : 6.283299
Lower : 201701-3529675195 : 6.193112
Higher : 201702-3569774772 : 6.145684
 : 201702-3578458483 : 6.112609
 : 201701-3554601981 : 6.051448
 : 201702-3571219128 : 5.976727
 : 201702-3568933298 : 5.909887
 : 201702-3568044996 : 5.903648
 : 201702-3561287800 : 5.849408
 : 201701-3530845829 : 5.803005
 : 201701-3552839669 : 5.7624
 : 201701-3553338968 : 5.737921
 : 201702-3583897710 : 5.732034
 : 201701-3552839678 : 5.653676
 : 201702-3566576918 : 5.575426
 : 201701-3532437024 : 5.566789
 : 201702-3581860923 : 5.530982
 : 201701-3532466453 : 5.480655
 : 201701-3576425286 : 5.40565
 : 201701-3563328422 : 5.401547
 : 201701-3572143522 : 5.385685
 : 201701-3546436350 : 5.384069
 : 201702-3583180142 : 5.376196

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Phrase: auto burglary Boost: 1
Term: cash Boost: 1
Term: atm Boost: 1
Term: jugging Boost: 1
Term: stolen Boost: 1
Term: car Boost: 1
Term: vehicle Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201701-3553047313 : 6.017861
Lower : 201701-3553159764 : 5.043849
Lower : 201701-3555037090 : 5.039083
Higher : 201702-3576532375 : 5.032813
Higher : 201701-3558936852 : 5.028587
Lower : 201702-3569876106 : 5.027058
Lower : 201702-3576425420 : 5.024188
Just Right : 201701-3534467738 : 5.022985
Lower : 201702-3564994922 : 5.022969
Lower : 201702-3578258881 : 5.022272
 : 201701-3581648839 : 5.021875
 : 201701-3572143496 : 5.02167
 : 201701-3566724094 : 5.021509
 : 201701-3547399516 : 5.020657
 : 201701-3530092154 : 5.020207
 : 201701-3553298086 : 5.01932
 : 201701-3529411642 : 5.018902
 : 201701-3528767276 : 5.017861
 : 201701-3575020597 : 5.017856
 : 201701-3534467831 : 5.017682
 : 201701-3530852809 : 5.017507
 : 201702-3581648879 : 5.0175
 : 201702-3575573711 : 5.017012
 : 201702-3572259664 : 5.016644
 : 201701-3558936849 : 5.016428
 : 201701-3570115254 : 5.01464
 : 201701-3539538991 : 5.014289
 : 201701-3546441154 : 5.014121
 : 201701-3546604804 : 5.013352
 : 201702-3560991337 : 5.01263

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Phrase: white male Boost: 1
Phrase: red hair Boost: 1
Phrase: red beard Boost: 1
Phrase: blue harley Boost: 1
Phrase: blue harley davidson Boost: 1
Phrase: blue motorcycle Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201701-3555053432 : 3.022332
Higher : 201701-3534344292 : 3.015657
Lower : 201701-3534467756 : 3.013983
Higher : 201701-3560147488 : 3.011513
Much Higher : 201701-3537818541 : 3.001288
Lower : 201702-3575296581 : 3.001113
Much Higher : 201701-3553351184 : 3.00103
Lower : 201702-3574533865 : 3.000954
Higher : 201701-3537527900 : 3.000795
Higher : 201701-3528767150 : 3.000674
 : 201701-3530371215 : 2.037349
 : 201702-3574533928 : 2.019568
 : 201701-3561085606 : 2.018078
 : 201701-3559022722 : 2.017209
 : 201702-3573182081 : 2.011707
 : 201702-3576718799 : 2.009425
 : 201701-3534083691 : 2.008502
 : 201702-3576425216 : 2.008396
 : 201701-3569692612 : 2.007915
 : 201702-3578456910 : 2.007358
 : 201702-3572268292 : 2.007
 : 201702-3575296504 : 2.006833
 : 201701-3555037902 : 2.006806
 : 201702-3580278201 : 2.006615
 : 201701-3539538968 : 2.006169
 : 201701-3558931268 : 2.005916
 : 201701-3533920235 : 2.005739
 : 201702-3575788263 : 2.005709
 : 201702-3571613427 : 2.005539
 : 201701-3568950654 : 2.005518

Score Type: Analysis Score Form A Analysis Number 1
```

```
Query Detail:
Phrase: auto burglary Boost: 4
Term: cash Boost: 4
Term: atm Boost: 1
Term: jugging Boost: 4
Term: stolen Boost: 1
Term: car Boost: 1
Term: vehicle Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Much Lower : 201702-3568398555 : 27.9412
Much Lower : 201702-3574765362 : 22.40427
Just Right : 201701-3552295848 : 21.46362
Higher : 201702-3572895997 : 20.26794
Lower : 201701-3531883026 : 20.04505
Lower : 201701-3538258211 : 20.02976
Higher : 201701-3553159764 : 19.8784
Just Right : 201701-3532603252 : 18.61814
Just Right : 201702-3574765354 : 18.49165
Lower : 201701-3531052688 : 18.45556
 : 201701-3559179448 : 18.33543
 : 201701-3554806870 : 18.21107
 : 201702-3581624389 : 18.13889
 : 201701-3555461524 : 18.06799
 : 201701-3583599210 : 17.89837
 : 201702-3582769291 : 17.85221
 : 201702-3582357315 : 17.78942
 : 201701-3555037090 : 17.64467
 : 201701-3538562158 : 17.56438
 : 201701-3551043556 : 17.50673
 : 201702-3576532375 : 17.50634
 : 201701-3528984766 : 17.4707
 : 201702-3570115455 : 17.42706
 : 201701-3538562491 : 17.34909
 : 201701-3554085744 : 17.34222
 : 201701-3551675766 : 17.26505
 : 201701-3582343497 : 17.25286
 : 201702-3578074510 : 17.16836
 : 201701-3531882989 : 17.11268
 : 201701-3546680495 : 17.09837

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Phrase: auto burglary Boost: 4
Term: cash Boost: 4
Term: atm Boost: 1
Term: jugging Boost: 4
Term: stolen Boost: 1
Term: car Boost: 1
Term: vehicle Boost: 1
```

```
The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201701-3553047313 : 6.017861
Just Right : 201701-3553159764 : 5.043849
Just Right : 201701-3555037090 : 5.039083
Just Right : 201702-3576532375 : 5.032813
Just Right : 201702-3569876106 : 5.027058
Just Right : 201701-3534467738 : 5.022985
Lower : 201702-3564994922 : 5.022969
Higher : 201701-3581648839 : 5.021875
Just Right : 201701-3572143496 : 5.02167
Lower : 201701-3547399516 : 5.020657
 : 201701-3530092154 : 5.020207
 : 201701-3553298086 : 5.01932
 : 201701-3529411642 : 5.018902
 : 201701-3528767276 : 5.017861
 : 201701-3534467831 : 5.017682
 : 201702-3581648879 : 5.0175
 : 201701-3558936849 : 5.016428
 : 201701-3570115254 : 5.01464
 : 201701-3539538991 : 5.014289
 : 201701-3546441154 : 5.014121
 : 201701-3558936852 : 5.012978
 : 201702-3560991337 : 5.01263
 : 201701-3566460817 : 5.010938
 : 201701-3543846346 : 5.010828
 : 201702-3581649129 : 5.009585
 : 201702-3576425420 : 5.009279
 : 201701-3530852684 : 5.008931
 : 201702-3578258881 : 5.008714
 : 201701-3566724094 : 5.008499
 : 201701-3530852809 : 5.007718

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Term: apartment Boost: 1
Term: burglaries Boost: 1
Term: firearms Boost: 4
Term: stolen Boost: 1
Term: taken Boost: 1
Phrase: window entry Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Much Lower : 201701-3531895240 : 22.81078
Much Lower : 201701-3533513345 : 19.35559
Lower : 201702-3574754867 : 18.88259
Lower : 201701-3551971493 : 18.72512
Just Right : 201702-3575442715 : 18.17107
Lower : 201702-3581571737 : 18.12507
Lower : 201701-3530613251 : 17.45535
Much Lower : 201701-3537607129 : 17.38905
Just Right : 201701-3533649312 : 17.23791
Just Right : 201701-3555311664 : 17.2149
 : 201701-3537520939 : 16.92212
 : 201702-3582378258 : 16.74322
 : 201702-3571380612 : 16.71756
 : 201701-3561144663 : 16.66289
 : 201701-3532451863 : 16.55784
 : 201702-3581879725 : 16.3528
 : 201701-3568272726 : 16.08308
 : 201702-3583205172 : 15.65989
 : 201701-3564994815 : 15.46645
 : 201702-3576718440 : 15.18163
 : 201701-3551971500 : 14.98152
 : 201702-3564995192 : 14.96253
 : 201702-3566458655 : 14.85114
 : 201701-3530092161 : 14.63742
 : 201701-3547394789 : 14.57264
 : 201701-3576425256 : 14.51001
 : 201702-3581648895 : 14.50769
 : 201701-3531683316 : 14.48811
 : 201701-3569876405 : 14.40767
 : 201701-3537527910 : 14.36652

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Term: apartment Boost: 1
Term: burglaries Boost: 1
Term: firearms Boost: 4
Term: stolen Boost: 1
Term: taken Boost: 1
Phrase: window entry Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201701-3532451863 : 5.019366
Just Right : 201701-3531683316 : 5.016945
Just Right : 201701-3546548159 : 5.014791
Just Right : 201701-3547394758 : 5.013073
Just Right : 201702-3563752049 : 5.008874
Just Right : 201702-3580875137 : 5.00882
Just Right : 201702-3560789830 : 5.008461
Just Right : 201702-3562562680 : 5.007977
Just Right : 201701-3563752902 : 5.007395
Just Right : 201702-3563752703 : 5.00701
 : 201702-3568950008 : 5.006008
 : 201702-3580781990 : 5.005983
 : 201701-3580781614 : 5.005847
 : 201701-3566568263 : 4.02371
 : 201701-3532466348 : 4.022224
 : 201701-3580316069 : 4.021647
 : 201701-3576526916 : 4.021535
 : 201701-3565597405 : 4.020957
 : 201701-3565596078 : 4.020122
 : 201701-3559034511 : 4.019923
 : 201701-3581882017 : 4.019587
 : 201701-3575295993 : 4.018093
 : 201701-3534467776 : 4.01796
 : 201702-3569235403 : 4.016482
 : 201702-3584008996 : 4.015536
 : 201701-3530935200 : 4.015276
 : 201701-3566567422 : 4.014785
 : 201701-3574765774 : 4.014289
 : 201702-3583113334 : 4.014271
 : 201701-3565596189 : 4.013939

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Term: abandoned Boost: 4
Term: homes Boost: 1
Term: buildings Boost: 1
Term: drugs Boost: 2
Term: homeless Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201701-3551358691 : 21.05244
Lower : 201701-3558931270 : 18.28111
Much Lower : 201702-3578333605 : 18.08944
Lower : 201701-3538063287 : 15.14274
Higher : 201702-3583178342 : 13.87002
Much Lower : 201701-3530287171 : 13.70767
Lower : 201702-3582060042 : 13.01538
Just Right : 201701-3529411666 : 12.23718
Lower : 201701-3539536396 : 11.99687
Much Lower : 201702-3572423205 : 11.9873
 : 201701-3538277758 : 11.96359
 : 201701-3530213562 : 11.74943
 : 201702-3577567442 : 11.67941
 : 201701-3534532514 : 11.54548
 : 201701-3551971608 : 11.39497
 : 201702-3574766182 : 11.27165
 : 201701-3529540393 : 11.19227
 : 201702-3561086240 : 11.19153
 : 201701-3554759712 : 11.14858
 : 201702-3562877936 : 10.99085
 : 201701-3561091695 : 10.95301
 : 201701-3538062546 : 10.92864
 : 201701-3542960376 : 10.81625
 : 201701-3553159753 : 10.38794
 : 201702-3562745461 : 10.38047
 : 201701-3529480561 : 10.32659
 : 201701-3530371197 : 10.11796
 : 201702-3584307275 : 10.08473
 : 201701-3551743293 : 9.89819
 : 201701-3540800671 : 9.772079

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Term: abandoned Boost: 4
Term: homes Boost: 1
Term: buildings Boost: 1
Term: drugs Boost: 2
Term: homeless Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201702-3583178342 : 3.056816
Just Right : 201702-3562745461 : 3.042521
Just Right : 201702-3581866240 : 3.03515
Just Right : 201702-3579619553 : 3.031818
Lower : 201701-3551358691 : 2.057492
Higher : 201701-3558931270 : 2.049923
Just Right : 201701-3538063287 : 2.041353
Lower : 201702-3582060042 : 2.035543
Higher : 201701-3529411666 : 2.033418
Lower : 201701-3539536396 : 2.032762
 : 201701-3538277758 : 2.032671
 : 201702-3577567442 : 2.031895
 : 201701-3551971608 : 2.031118
 : 201702-3574766182 : 2.030782
 : 201702-3561086240 : 2.030563
 : 201701-3554759712 : 2.030445
 : 201702-3562877936 : 2.030015
 : 201701-3542960376 : 2.029538
 : 201701-3553159753 : 2.028368
 : 201702-3584307275 : 2.02754
 : 201701-3544913914 : 2.026351
 : 201701-3547399628 : 2.02621
 : 201701-3543819748 : 2.026108
 : 201701-3552145007 : 2.024676
 : 201702-3578840748 : 2.024356
 : 201701-3531471122 : 2.02428
 : 201701-3532881581 : 2.024256
 : 201701-3544913908 : 2.024103
 : 201701-3560142953 : 2.023955
 : 201701-3537526986 : 2.02363

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Phrase: white male Boost: 2
Phrase: red hair Boost: 2
Phrase: red beard Boost: 2
Phrase: blue harley Boost: 2
```

```
Phrase: blue harley davidson Boost: 1
Phrase: blue motorcycle Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201701-3530380480 : 46.03281
Much Lower : 201701-3530371215 : 37.11855
Just Right : 201701-3530845632 : 27.49674
Just Right : 201701-3532358245 : 16.27506
Just Right : 201701-3553624971 : 15.34117
Higher : 201702-3580755972 : 14.04208
Just Right : 201701-3555053432 : 13.73276
Just Right : 201702-3575296581 : 13.70869
Lower : 201701-3531613188 : 12.64299
Lower : 201701-3531409708 : 12.2655
Higher : 201701-3534344292 : 12.13698
Higher : 201701-3537818541 : 12.01941
Higher : 201701-3528767150 : 10.90795
Much Lower : 201701-3534083686 : 10.84127
 : 201702-3578270081 : 10.82464
 : 201702-3561091676 : 10.78693
 : 201701-3547595305 : 10.53156
 : 201701-3561085606 : 10.50004
 : 201701-3537527900 : 10.21261
 : 201701-3559022722 : 10.20742
 : 201702-3580322447 : 10.15556
 : 201701-3530908912 : 9.696728
 : 201701-3553351184 : 9.615526
 : 201702-3581882064 : 9.126972
 : 201701-3582060456 : 9.120598
 : 201702-3583429511 : 8.886371
 : 201701-3551971645 : 8.694165
 : 201702-3583429525 : 8.659714
 : 201701-3530935310 : 8.570777
 : 201701-3530611211 : 8.43084

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Phrase: white male Boost: 2
Phrase: red hair Boost: 2
Phrase: red beard Boost: 2
Phrase: blue harley Boost: 2
Phrase: blue harley davidson Boost: 1
Phrase: blue motorcycle Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Much Lower : 201701-3555053432 : 3.014144
Just Right : 201701-3534344292 : 3.011106
Just Right : 201701-3534467756 : 3.009918
Just Right : 201701-3560147488 : 3.008167
Just Right : 201701-3537818541 : 3.000913
Just Right : 201702-3575296581 : 3.000789
Just Right : 201701-3553351184 : 3.000731
Just Right : 201702-3574533865 : 3.000676
Just Right : 201701-3537527900 : 3.000564
Just Right : 201701-3528767150 : 3.000478
 : 201701-3530371215 : 2.022737
 : 201702-3574533928 : 2.018553
 : 201701-3561085606 : 2.01714
 : 201701-3559022722 : 2.016316
 : 201702-3573182081 : 2.0111
 : 201702-3576718799 : 2.008936
 : 201701-3534083691 : 2.008061
 : 201702-3576425216 : 2.00796
 : 201701-3569692612 : 2.007505
 : 201702-3578456910 : 2.006976
 : 201702-3572268292 : 2.006637
 : 201702-3575296504 : 2.006478
 : 201701-3555037902 : 2.006453
 : 201702-3580278201 : 2.006272
 : 201701-3539538968 : 2.005849
 : 201701-3533920235 : 2.005441
 : 201702-3575788263 : 2.005413
 : 201702-3571613427 : 2.005251
 : 201701-3568950654 : 2.005232
 : 201701-3551971389 : 2.005065

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Term: hispanic Boost: 4
Term: latina Boost: 1
Term: latin Boost: 1
Term: female Boost: 1
Phrase: h f Boost: 1
Phrase: l f Boost: 1
Term: light Boost: 2
Term: blonde Boost: 2
Term: male Boost: 4
Term: thin Boost: 1
Term: skinny Boost: 2
Term: committing Boost: 1
Term: thefts Boost: 1
Term: stealing Boost: 1
Term: shoplifting Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201702-3562752881 : 23.18534
Lower : 201702-3566728881 : 19.24125
Lower : 201701-3552839669 : 19.12174
Lower : 201701-3563328422 : 17.75004
Lower : 201701-3552839678 : 17.70141
Lower : 201701-3552562924 : 17.56799
Much Lower : 201702-3582358264 : 17.48615
Much Higher : 201701-3533272388 : 17.4697
Higher : 201701-3537370154 : 17.44232
Much Higher : 201701-3532466453 : 17.32256
Lower : 201702-3578270059 : 17.05994
 : 201702-3579365014 : 17.00743
 : 201701-3551019435 : 16.77782
 : 201701-3562878015 : 16.73153
 : 201702-3576647914 : 15.82072
 : 201701-3530730084 : 15.7472
 : 201702-3571219128 : 15.57203
 : 201701-3553824290 : 15.57183
 : 201701-3574765772 : 15.50855
 : 201701-3544454109 : 15.47671
 : 201702-3568044996 : 15.27971
 : 201701-3530845829 : 15.15901
 : 201701-3553157739 : 15.14866
 : 201702-3569774772 : 14.95835
 : 201701-3552088809 : 14.82568
 : 201702-3583216808 : 14.59839
 : 201702-3583537972 : 14.53002
 : 201701-3531683305 : 14.37915
 : 201702-3580755991 : 14.10345
 : 201701-3555230079 : 14.09763

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Term: hispanic Boost: 4
Term: latina Boost: 1
Term: latin Boost: 1
Term: female Boost: 1
Phrase: h f Boost: 1
Phrase: l f Boost: 1
Term: light Boost: 2
Term: blonde Boost: 2
Term: male Boost: 4
Term: thin Boost: 1
Term: skinny Boost: 2
Term: committing Boost: 1
Term: thefts Boost: 1
Term: stealing Boost: 1
Term: shoplifting Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201701-3553641699 : 8.011668
Lower : 201701-3551757787 : 8.011173
Lower : 201701-3560789631 : 8.010921
Lower : 201701-3581567209 : 8.009311
Higher : 201702-3579142671 : 7.011557
Lower : 201702-3572268329 : 7.0029
```

```
Much Lower : 201702-3581550107 : 7.00287
Just Right : 201701-3560086702 : 6.022195
Just Right : 201701-3559044296 : 6.009092
Lower : 201701-3547595233 : 6.007138
 : 201702-3578964639 : 6.007049
 : 201702-3579633198 : 6.004963
 : 201702-3581106516 : 6.004877
 : 201702-3574855935 : 6.004841
 : 201702-3581550093 : 6.004827
 : 201702-3577525241 : 6.004812
 : 201702-3575579555 : 6.004798
 : 201702-3576532398 : 6.004774
 : 201702-3569674850 : 6.004555
 : 201702-3580322486 : 6.00418
 : 201701-3560985505 : 6.003694
 : 201701-3547314580 : 6.003656
 : 201701-3547314704 : 6.003134
 : 201701-3552839669 : 5.014605
 : 201701-3552839678 : 5.013503
 : 201701-3532466453 : 5.013438
 : 201701-3562878015 : 5.012779
 : 201702-3571219128 : 5.01208
 : 201702-3568044996 : 5.011853
 : 201701-3574765772 : 5.01183

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Phrase: black male Boost: 4
Phrase: african american male Boost: 1
Phrase: sleeve tattoos Boost: 4
Phrase: arm tattoos Boost: 4
Phrase: black chevrolet impala Boost: 2
Phrase: chevy impala Boost: 1
Phrase: black impala Boost: 1
Phrase: chevy sedan Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201701-3551971669 : 40.01641
Much Lower : 201701-3575610634 : 36.17804
Lower : 201701-3554543133 : 32.07912
Lower : 201701-3532603484 : 31.7567
Much Higher : 201702-3574755931 : 30.97523
Lower : 201701-3530145912 : 30.87152
Lower : 201701-3562867902 : 30.54404
Just Right : 201702-3582480497 : 29.75882
Much Lower : 201701-3554370121 : 29.20919
Just Right : 201702-3584530034 : 26.97574
 : 201702-3568028249 : 25.88299
 : 201701-3531882708 : 25.66329
 : 201702-3581648950 : 25.58174
 : 201701-3530084891 : 25.43297
 : 201701-3530072922 : 25.31185
 : 201701-3531613176 : 24.9391
 : 201702-3568028310 : 24.3753
 : 201702-3565649617 : 23.51281
 : 201701-3543846384 : 23.35283
 : 201701-3548054371 : 22.43403
 : 201701-3551963881 : 22.14326
 : 201702-3568269959 : 21.69587
 : 201701-3539348024 : 21.45271
 : 201701-3532881633 : 21.31812
 : 201702-3568933846 : 21.11821
 : 201701-3548165189 : 20.78259
 : 201701-3530092155 : 20.59526
 : 201701-3533064191 : 20.57371
 : 201702-3582480506 : 20.56869
 : 201701-3529401838 : 20.45506

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Phrase: black male Boost: 4
Phrase: african american male Boost: 1
Phrase: sleeve tattoos Boost: 4
Phrase: arm tattoos Boost: 4
Phrase: black chevrolet impala Boost: 2
Phrase: chevy impala Boost: 1
Phrase: black impala Boost: 1
Phrase: chevy sedan Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201702-3568028310 : 4.032466
Just Right : 201702-3580782744 : 4.021355
Just Right : 201702-3568028249 : 4.000709
Just Right : 201702-3583429481 : 3.016466
Just Right : 201702-3566456690 : 3.010837
Just Right : 201701-3546441267 : 3.008538
Just Right : 201701-3575442527 : 3.00805
Just Right : 201701-3577404468 : 3.007871
Much Higher : 201702-3574755931 : 3.001219
Just Right : 201701-3551971669 : 3.000964
 : 201702-3572143546 : 3.000669
 : 201701-3530084891 : 3.000563
 : 201702-3571380314 : 3.000493
 : 201702-3572143783 : 3.000427
 : 201702-3569359435 : 3.000427
 : 201702-3582480497 : 2.018611
 : 201701-3562867902 : 2.01813
 : 201701-3531613176 : 2.016715
 : 201702-3565649617 : 2.015759
 : 201702-3572248011 : 2.015409
 : 201702-3580076490 : 2.01425
 : 201701-3533064191 : 2.013789
 : 201701-3532358243 : 2.013257
 : 201701-3532603484 : 2.012866
 : 201702-3581881996 : 2.011942
 : 201701-3554543133 : 2.011508
 : 201701-3531882708 : 2.010397
 : 201702-3578270001 : 2.009931
 : 201701-3555037064 : 2.008773
 : 201701-3569977157 : 2.008677

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Phrase: white male Boost: 2
Phrase: caucasian male Boost: 1
Term: wm Boost: 2
Term: kkk Boost: 1
Phrase: white supremacy Boost: 1
Phrase: neo nazi Boost: 1
Term: ryan Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201701-3533533466 : 22.02452
Lower : 201701-3581736460 : 20.50029
Just Right : 201701-3581571429 : 19.37165
Just Right : 201701-3553645402 : 18.64038
Lower : 201702-3560924119 : 16.73842
Lower : 201701-3537451662 : 15.78113
Just Right : 201701-3555036327 : 14.9123
Higher : 201702-3561091607 : 14.84971
Lower : 201701-3554385548 : 14.79481
Higher : 201701-3538060958 : 14.36074
 : 201701-3553641891 : 14.23346
 : 201701-3533528729 : 13.3638
 : 201701-3547086938 : 13.23288
 : 201701-3538496722 : 13.09307
 : 201701-3552304802 : 13.0478
 : 201701-3530852856 : 12.84585
 : 201702-3583537911 : 12.76729
 : 201702-3578385967 : 12.6503
 : 201701-3531613188 : 12.64299
 : 201701-3531409708 : 12.2655
 : 201702-3573706320 : 12.1855
 : 201701-3531699376 : 12.07991
 : 201701-3530380078 : 11.67193
 : 201702-3581997465 : 11.33893
 : 201701-3555053602 : 11.31934
 : 201701-3530610680 : 11.18541
 : 201701-3569969826 : 11.18423
 : 201702-3570190760 : 11.07299
 : 201701-3551971462 : 11.07142
 : 201701-3561144591 : 11.0274
```

```
Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Phrase: white male Boost: 2
Phrase: caucasian male Boost: 1
Term: wm Boost: 2
Term: kkk Boost: 1
Phrase: white supremacy Boost: 1
Phrase: neo nazi Boost: 1
Term: ryan Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201701-3565732115 : 3.002517
Higher : 201701-3533533466 : 2.045767
Just Right : 201701-3533528729 : 2.02777
Just Right : 201701-3538496722 : 2.027207
Just Right : 201701-3530852856 : 2.026694
Lower : 201702-3583537911 : 2.02653
Just Right : 201702-3578385967 : 2.026287
Just Right : 201701-3530380078 : 2.024254
Just Right : 201702-3581997465 : 2.023562
Lower : 201701-3555053602 : 2.023521
 : 201702-3582641547 : 2.020958
 : 201701-3555036184 : 2.02079
 : 201701-3553127386 : 2.020534
 : 201701-3558684975 : 2.020406
 : 201701-3530611179 : 2.019238
 : 201701-3530629543 : 2.019067
 : 201702-3572143627 : 2.019056
 : 201701-3542747825 : 2.018971
 : 201701-3553971032 : 2.018862
 : 201701-3558702085 : 2.018717
 : 201702-3581567726 : 2.018425
 : 201701-3552304792 : 2.017672
 : 201701-3555037660 : 2.017525
 : 201701-3538063031 : 2.017465
 : 201701-3528925183 : 2.017004
 : 201702-3569977136 : 2.016799
 : 201701-3554385475 : 2.016179
 : 201701-3561203382 : 2.01572
 : 201702-3570433440 : 2.015523
 : 201701-3528995178 : 2.015317

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Phrase: auto burglaries Boost: 2
Phrase: vehicle burglary Boost: 2
Phrase: grand theft Boost: 1
Phrase: grand theft auto Boost: 1
Term: cash Boost: 1
Term: bank Boost: 6
```

```
Term: jugging Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Much Lower : 201702-3579639343 : 26.25785
Much Lower : 201702-3581881822 : 25.06845
Much Lower : 201701-3532353801 : 24.86535
Much Lower : 201701-3551210097 : 23.42274
Much Lower : 201701-3528699223 : 22.24697
Lower : 201701-3568933395 : 21.39086
Much Lower : 201701-3562878013 : 21.02579
Much Lower : 201702-3574856047 : 20.99512
Lower : 201701-3554541020 : 20.5533
Lower : 201701-3565607684 : 20.11711
 : 201702-3574753751 : 20.09968
 : 201701-3561091611 : 19.83361
 : 201701-3557692596 : 19.4358
 : 201702-3576105611 : 19.40315
 : 201701-3569977177 : 19.27958
 : 201702-3575579526 : 18.86583
 : 201702-3569971943 : 18.46624
 : 201702-3581881749 : 18.39373
 : 201701-3560140796 : 18.02779
 : 201701-3547437911 : 18.02628
 : 201701-3546231890 : 18.0221
 : 201701-3553624995 : 17.98203
 : 201701-3537320893 : 17.95075
 : 201701-3546551842 : 17.63919
 : 201702-3578430740 : 17.58722
 : 201702-3576718381 : 17.56901
 : 201701-3580020809 : 17.42725
 : 201701-3539655573 : 17.20041
 : 201702-3580778829 : 17.17641
 : 201702-3574755112 : 17.1742

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Phrase: auto burglaries Boost: 2
Phrase: vehicle burglary Boost: 2
Phrase: grand theft Boost: 1
Phrase: grand theft auto Boost: 1
Term: cash Boost: 1
Term: bank Boost: 6
Term: jugging Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201701-3580020809 : 5.022854
Just Right : 201701-3571201305 : 5.020456
Just Right : 201702-3582374538 : 5.01336
Just Right : 201702-3581567347 : 5.01157
```

```
Lower : 201701-3537320893 : 4.021947
Lower : 201701-3547437911 : 4.016285
Higher : 201701-3544915216 : 4.016052
Just Right : 201702-3570115318 : 4.011815
Just Right : 201702-3571613414 : 4.01057
Just Right : 201702-3579681363 : 4.010109
 : 201702-3574906360 : 4.009839
 : 201701-3566568263 : 4.00649
 : 201702-3579639343 : 3.024078
 : 201702-3581881822 : 3.022987
 : 201701-3532353801 : 3.022801
 : 201701-3554541020 : 3.018847
 : 201702-3576105611 : 3.017792
 : 201702-3575579526 : 3.017299
 : 201702-3569971943 : 3.016933
 : 201702-3565603181 : 3.015642
 : 201702-3578258881 : 3.015008
 : 201702-3581550850 : 3.014355
 : 201702-3572143817 : 3.013917
 : 201702-3567918668 : 3.013914
 : 201701-3548054504 : 3.01377
 : 201701-3560991379 : 3.013617
 : 201701-3553824229 : 3.012991
 : 201702-3572259664 : 3.012869
 : 201702-3580756797 : 3.012609
 : 201702-3582358262 : 3.012225

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Term: burglaries Boost: 1
Phrase: break in Boost: 1
Phrase: forcible entry Boost: 1
Phrase: roof entry Boost: 4
Phrase: business burglary Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201701-3581648849 : 33.65192
Just Right : 201701-3560000309 : 29.43406
Much Lower : 201701-3544602378 : 24.08854
Lower : 201702-3573648162 : 21.54545
Lower : 201701-3546441409 : 17.03317
Much Higher : 201701-3553047388 : 14.51171
Lower : 201701-3530935200 : 13.62654
Much Lower : 201702-3570049703 : 12.45412
Lower : 201701-3548066840 : 12.04427
Lower : 201702-3576588670 : 9.363263
 : 201702-3573648160 : 9.140507
 : 201701-3530908890 : 7.895797
 : 201701-3547438419 : 7.400162
 : 201702-3564904796 : 7.334181
 : 201701-3529469197 : 6.978964
 : 201701-3550356456 : 6.908822
 : 201701-3544491355 : 6.907141
 : 201701-3530144991 : 6.83796
 : 201702-3571470696 : 6.530951
 : 201702-3576584997 : 6.183417
 : 201702-3583898309 : 5.983215
 : 201701-3532530493 : 5.907507
 : 201701-3551292812 : 5.806771
 : 201702-3581298330 : 5.725419
 : 201701-3528827279 : 5.583171
 : 201701-3543221430 : 5.542486
 : 201702-3577404646 : 5.534036
 : 201701-3543192605 : 5.403693
 : 201701-3560000261 : 5.133927
 : 201701-3577404482 : 5.118783

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Term: burglaries Boost: 1
Phrase: break in Boost: 1
Phrase: forcible entry Boost: 1
Phrase: roof entry Boost: 4
Phrase: business burglary Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Much Lower : 201702-3577404646 : 3.000411
Lower : 201701-3563752671 : 3.000318
Lower : 201701-3564994855 : 3.0003
Lower : 201701-3547438419 : 3.000265
Higher : 201701-3581648849 : 2.037809
Higher : 201701-3560000309 : 2.032886
Higher : 201701-3553047388 : 2.017176
Much Lower : 201702-3570049703 : 2.015279
Much Lower : 201701-3532530493 : 2.013741
Much Higher : 201702-3573648160 : 2.011456
 : 201701-3544491355 : 2.011196
 : 201702-3571470696 : 2.009008
 : 201702-3583113277 : 2.008097
 : 201701-3543192605 : 2.007699
 : 201701-3546441151 : 2.007698
 : 201701-3562752768 : 2.007081
 : 201701-3530212289 : 2.006644
 : 201702-3579681194 : 2.006571
 : 201702-3582641426 : 2.005632
 : 201701-3581648840 : 2.005178
 : 201702-3574765504 : 2.004721
 : 201701-3563752689 : 2.003071
 : 201701-3563752029 : 2.002193
 : 201701-3543221430 : 2.00053
 : 201702-3579617007 : 2.000433
 : 201701-3551971513 : 2.00039
 : 201701-3538294014 : 2.000375
 : 201701-3569359227 : 2.000371
 : 201702-3582060576 : 2.000325
 : 201702-3580078297 : 2.0003

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Phrase: construction site thefts Boost: 1
Phrase: construction burglaries Boost: 1
Phrase: stolen construction equipment Boost: 1
Phrase: new build neighborhoods Boost: 1
Phrase: construction thefts Boost: 1

The number of records found: 9
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201701-3572268274 : 24.43997
Just Right : 201701-3570432463 : 8.533347
Just Right : 201701-3566949683 : 7.542485
Just Right : 201701-3576983744 : 7.466679
Just Right : 201701-3582343657 : 7.390096
Just Right : 201701-3562562618 : 5.66007
Just Right : 201702-3580316330 : 5.213573
Just Right : 201701-3560985485 : 2.474575
Just Right : 201701-3547314543 : 2.449704

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Phrase: construction site thefts Boost: 2
Phrase: construction burglaries Boost: 2
Phrase: stolen construction equipment Boost: 2
Phrase: new build neighborhoods Boost: 1
Phrase: construction thefts Boost: 2

The number of records found: 9
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201701-3572268274 : 3.018437
Lower : 201701-3570432463 : 2.012541
Just Right : 201701-3566949683 : 2.011085
Just Right : 201701-3576983744 : 2.010973
Just Right : 201701-3582343657 : 2.010861
Just Right : 201702-3580316330 : 1.010257
Just Right : 201701-3562562618 : 1.004209
Higher : 201701-3560985485 : 1.00184
Higher : 201701-3547314543 : 1.001822

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Phrase: auto burglary Boost: 4
Term: cash Boost: 4
```

```
Term: atm Boost: 1
Term: jugging Boost: 4
Term: stolen Boost: 1
Term: car Boost: 1
Term: vehicle Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Much Lower : 201702-3568398555 : 27.9412
Much Lower : 201702-3574765362 : 22.40427
Just Right : 201701-3552295848 : 21.46362
Higher : 201702-3572895997 : 20.26794
Lower : 201701-3531883026 : 20.04505
Lower : 201701-3538258211 : 20.02976
Higher : 201701-3553159764 : 19.8784
Just Right : 201701-3532603252 : 18.61814
Just Right : 201702-3574765354 : 18.49165
Lower : 201701-3531052688 : 18.45556
 : 201701-3559179448 : 18.33543
 : 201701-3554806870 : 18.21107
 : 201702-3581624389 : 18.13889
 : 201701-3555461524 : 18.06799
 : 201701-3583599210 : 17.89837
 : 201702-3582769291 : 17.85221
 : 201702-3582357315 : 17.78942
 : 201701-3555037090 : 17.64467
 : 201701-3538562158 : 17.56438
 : 201701-3551043556 : 17.50673
 : 201702-3576532375 : 17.50634
 : 201701-3528984766 : 17.4707
 : 201702-3570115455 : 17.42706
 : 201701-3538562491 : 17.34909
 : 201701-3554085744 : 17.34222
 : 201701-3551675766 : 17.26505
 : 201701-3582343497 : 17.25286
 : 201702-3578074510 : 17.16836
 : 201701-3531882989 : 17.11268
 : 201701-3546680495 : 17.09837

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Phrase: auto burglary Boost: 4
Term: cash Boost: 4
Term: atm Boost: 1
Term: jugging Boost: 4
Term: stolen Boost: 1
Term: car Boost: 1
Term: vehicle Boost: 1

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
```

```
Just Right : 201701-3553047313 : 6.017861
Just Right : 201701-3553159764 : 5.043849
Just Right : 201701-3555037090 : 5.039083
Just Right : 201702-3576532375 : 5.032813
Just Right : 201702-3569876106 : 5.027058
Just Right : 201701-3534467738 : 5.022985
Lower : 201702-3564994922 : 5.022969
Higher : 201701-3581648839 : 5.021875
Just Right : 201701-3572143496 : 5.02167
Lower : 201701-3547399516 : 5.020657
 : 201701-3530092154 : 5.020207
 : 201701-3553298086 : 5.01932
 : 201701-3529411642 : 5.018902
 : 201701-3528767276 : 5.017861
 : 201701-3534467831 : 5.017682
 : 201702-3581648879 : 5.0175
 : 201701-3558936849 : 5.016428
 : 201701-3570115254 : 5.01464
 : 201701-3539538991 : 5.014289
 : 201701-3546441154 : 5.014121
 : 201701-3558936852 : 5.012978
 : 201702-3560991337 : 5.01263
 : 201701-3566460817 : 5.010938
 : 201701-3543846346 : 5.010828
 : 201702-3581649129 : 5.009585
 : 201702-3576425420 : 5.009279
 : 201701-3530852684 : 5.008931
 : 201702-3578258881 : 5.008714
 : 201701-3566724094 : 5.008499
 : 201701-3530852809 : 5.007718

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Term: apartment Boost: 1
Term: burglaries Boost: 1
Term: firearms Boost: 4
Term: stolen Boost: 1
Term: taken Boost: 1
Phrase: window entry Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Much Lower : 201701-3531895240 : 22.81078
Much Lower : 201701-3533513345 : 19.35559
Lower : 201702-3574754867 : 18.88259
Lower : 201701-3551971493 : 18.72512
Just Right : 201702-3575442715 : 18.17107
Lower : 201702-3581571737 : 18.12507
Lower : 201701-3530613251 : 17.45535
Much Lower : 201701-3537607129 : 17.38905
Just Right : 201701-3533649312 : 17.23791
Just Right : 201701-3555311664 : 17.2149
```

```
 : 201701-3537520939 : 16.92212
 : 201702-3582378258 : 16.74322
 : 201702-3571380612 : 16.71756
 : 201701-3561144663 : 16.66289
 : 201701-3532451863 : 16.55784
 : 201702-3581879725 : 16.3528
 : 201701-3568272726 : 16.08308
 : 201702-3583205172 : 15.65989
 : 201701-3564994815 : 15.46645
 : 201702-3576718440 : 15.18163
 : 201701-3551971500 : 14.98152
 : 201702-3564995192 : 14.96253
 : 201702-3566458655 : 14.85114
 : 201701-3530092161 : 14.63742
 : 201701-3547394789 : 14.57264
 : 201701-3576425256 : 14.51001
 : 201702-3581648895 : 14.50769
 : 201701-3531683316 : 14.48811
 : 201701-3569876405 : 14.40767
 : 201701-3537527910 : 14.36652

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Term: apartment Boost: 1
Term: burglaries Boost: 1
Term: firearms Boost: 4
Term: stolen Boost: 1
Term: taken Boost: 1
Phrase: window entry Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201701-3532451863 : 5.019366
Just Right : 201701-3531683316 : 5.016945
Just Right : 201701-3546548159 : 5.014791
Just Right : 201701-3547394758 : 5.013073
Just Right : 201702-3563752049 : 5.008874
Just Right : 201702-3580875137 : 5.00882
Just Right : 201702-3560789830 : 5.008461
Just Right : 201702-3562562680 : 5.007977
Just Right : 201701-3563752902 : 5.007395
Just Right : 201702-3563752703 : 5.00701
 : 201702-3568950008 : 5.006008
 : 201702-3580781990 : 5.005983
 : 201701-3580781614 : 5.005847
 : 201701-3566568263 : 4.02371
 : 201701-3532466348 : 4.022224
 : 201701-3580316069 : 4.021647
 : 201701-3576526916 : 4.021535
 : 201701-3565597405 : 4.020957
 : 201701-3565596078 : 4.020122
 : 201701-3559034511 : 4.019923
 : 201701-3581882017 : 4.019587
 : 201701-3575295993 : 4.018093
 : 201701-3534467776 : 4.01796
 : 201702-3569235403 : 4.016482
 : 201702-3584008996 : 4.015536
 : 201701-3530935200 : 4.015276
 : 201701-3566567422 : 4.014785
 : 201701-3574765774 : 4.014289
 : 201702-3583113334 : 4.014271
 : 201701-3565596189 : 4.013939

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Term: abandoned Boost: 4
Term: homes Boost: 1
Term: buildings Boost: 1
Term: drugs Boost: 2
Term: homeless Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201701-3551358691 : 21.05244
Lower : 201701-3558931270 : 18.28111
Much Lower : 201702-3578333605 : 18.08944
Lower : 201701-3538063287 : 15.14274
Higher : 201702-3583178342 : 13.87002
Much Lower : 201701-3530287171 : 13.70767
Lower : 201702-3582060042 : 13.01538
Just Right : 201701-3529411666 : 12.23718
Lower : 201701-3539536396 : 11.99687
Much Lower : 201702-3572423205 : 11.9873
 : 201701-3538277758 : 11.96359
 : 201701-3530213562 : 11.74943
 : 201702-3577567442 : 11.67941
 : 201701-3534532514 : 11.54548
 : 201701-3551971608 : 11.39497
 : 201702-3574766182 : 11.27165
 : 201701-3529540393 : 11.19227
 : 201702-3561086240 : 11.19153
 : 201701-3554759712 : 11.14858
 : 201702-3562877936 : 10.99085
 : 201701-3561091695 : 10.95301
 : 201701-3538062546 : 10.92864
 : 201701-3542960376 : 10.81625
 : 201701-3553159753 : 10.38794
 : 201702-3562745461 : 10.38047
 : 201701-3529480561 : 10.32659
 : 201701-3530371197 : 10.11796
 : 201702-3584307275 : 10.08473
 : 201701-3551743293 : 9.89819
 : 201701-3540800671 : 9.772079
```

```
Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Term: abandoned Boost: 4
Term: homes Boost: 1
Term: buildings Boost: 1
Term: drugs Boost: 2
Term: homeless Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Just Right : 201702-3583178342 : 3.056816
Just Right : 201702-3562745461 : 3.042521
Just Right : 201702-3581866240 : 3.03515
Just Right : 201702-3579619553 : 3.031818
Lower : 201701-3551358691 : 2.057492
Higher : 201701-3558931270 : 2.049923
Just Right : 201701-3538063287 : 2.041353
Lower : 201702-3582060042 : 2.035543
Higher : 201701-3529411666 : 2.033418
Lower : 201701-3539536396 : 2.032762
 : 201701-3538277758 : 2.032671
 : 201702-3577567442 : 2.031895
 : 201701-3551971608 : 2.031118
 : 201702-3574766182 : 2.030782
 : 201702-3561086240 : 2.030563
 : 201701-3554759712 : 2.030445
 : 201702-3562877936 : 2.030015
 : 201701-3542960376 : 2.029538
 : 201701-3553159753 : 2.028368
 : 201702-3584307275 : 2.02754
 : 201701-3544913914 : 2.026351
 : 201701-3547399628 : 2.02621
 : 201701-3543819748 : 2.026108
 : 201701-3552145007 : 2.024676
 : 201702-3578840748 : 2.024356
 : 201701-3531471122 : 2.02428
 : 201701-3532881581 : 2.024256
 : 201701-3544913908 : 2.024103
 : 201701-3560142953 : 2.023955
 : 201701-3537526986 : 2.02363

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Phrase: white male Boost: 2
Phrase: red hair Boost: 2
Phrase: red beard Boost: 2
Phrase: blue harley Boost: 2
Phrase: blue harley davidson Boost: 1
Phrase: blue motorcycle Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201701-3530380480 : 46.03281
Much Lower : 201701-3530371215 : 37.11855
Just Right : 201701-3530845632 : 27.49674
Just Right : 201701-3532358245 : 16.27506
Just Right : 201701-3553624971 : 15.34117
Higher : 201702-3580755972 : 14.04208
Just Right : 201701-3555053432 : 13.73276
Just Right : 201702-3575296581 : 13.70869
Lower : 201701-3531613188 : 12.64299
Lower : 201701-3531409708 : 12.2655
Higher : 201701-3534344292 : 12.13698
Higher : 201701-3537818541 : 12.01941
Higher : 201701-3528767150 : 10.90795
Much Lower : 201701-3534083686 : 10.84127
 : 201702-3578270081 : 10.82464
 : 201702-3561091676 : 10.78693
 : 201701-3547595305 : 10.53156
 : 201701-3561085606 : 10.50004
 : 201701-3537527900 : 10.21261
 : 201701-3559022722 : 10.20742
 : 201702-3580322447 : 10.15556
 : 201701-3530908912 : 9.696728
 : 201701-3553351184 : 9.615526
 : 201702-3581882064 : 9.126972
 : 201701-3582060456 : 9.120598
 : 201702-3583429511 : 8.886371
 : 201701-3551971645 : 8.694165
 : 201702-3583429525 : 8.659714
 : 201701-3530935310 : 8.570777
 : 201701-3530611211 : 8.43084

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Phrase: white male Boost: 2
Phrase: red hair Boost: 2
Phrase: red beard Boost: 2
Phrase: blue harley Boost: 2
Phrase: blue harley davidson Boost: 1
Phrase: blue motorcycle Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Much Lower : 201701-3555053432 : 3.014144
Just Right : 201701-3534344292 : 3.011106
Just Right : 201701-3534467756 : 3.009918
Just Right : 201701-3560147488 : 3.008167
Just Right : 201701-3537818541 : 3.000913
Just Right : 201702-3575296581 : 3.000789
Just Right : 201701-3553351184 : 3.000731
Just Right : 201702-3574533865 : 3.000676
Just Right : 201701-3537527900 : 3.000564
Just Right : 201701-3528767150 : 3.000478
 : 201701-3530371215 : 2.022737
 : 201702-3574533928 : 2.018553
 : 201701-3561085606 : 2.01714
 : 201701-3559022722 : 2.016316
 : 201702-3573182081 : 2.0111
 : 201702-3576718799 : 2.008936
 : 201701-3534083691 : 2.008061
 : 201702-3576425216 : 2.00796
 : 201701-3569692612 : 2.007505
 : 201702-3578456910 : 2.006976
 : 201702-3572268292 : 2.006637
 : 201702-3575296504 : 2.006478
 : 201701-3555037902 : 2.006453
 : 201702-3580278201 : 2.006272
 : 201701-3539538968 : 2.005849
 : 201701-3533920235 : 2.005441
 : 201702-3575788263 : 2.005413
 : 201702-3571613427 : 2.005251
 : 201701-3568950654 : 2.005232
 : 201701-3551971389 : 2.005065

Score Type: Analysis Score Form A Analysis Number 1

Query Detail:
Term: hispanic Boost: 4
Term: latina Boost: 1
Term: latin Boost: 1
Term: female Boost: 1
Phrase: h f Boost: 1
Phrase: l f Boost: 1
Term: light Boost: 2
Term: blonde Boost: 2
Term: male Boost: 4
Term: thin Boost: 1
Term: skinny Boost: 2
Term: committing Boost: 1
Term: thefts Boost: 1
Term: stealing Boost: 1
Term: shoplifting Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201702-3562752881 : 23.18534
Lower : 201702-3566728881 : 19.24125
Lower : 201701-3552839669 : 19.12174
Lower : 201701-3563328422 : 17.75004
Lower : 201701-3552839678 : 17.70141
Lower : 201701-3552562924 : 17.56799
Much Lower : 201702-3582358264 : 17.48615
Much Higher : 201701-3533272388 : 17.4697
Higher : 201701-3537370154 : 17.44232
```

```
Much Higher : 201701-3532466453 : 17.32256
Lower : 201702-3578270059 : 17.05994
 : 201702-3579365014 : 17.00743
 : 201701-3551019435 : 16.77782
 : 201701-3562878015 : 16.73153
 : 201702-3576647914 : 15.82072
 : 201701-3530730084 : 15.7472
 : 201702-3571219128 : 15.57203
 : 201701-3553824290 : 15.57183
 : 201701-3574765772 : 15.50855
 : 201701-3544454109 : 15.47671
 : 201702-3568044996 : 15.27971
 : 201701-3530845829 : 15.15901
 : 201701-3553157739 : 15.14866
 : 201702-3569774772 : 14.95835
 : 201701-3552088809 : 14.82568
 : 201702-3583216808 : 14.59839
 : 201702-3583537972 : 14.53002
 : 201701-3531683305 : 14.37915
 : 201702-3580755991 : 14.10345
 : 201701-3555230079 : 14.09763

Score Type: Analysis Score Form B Analysis Number 1

Query Detail:
Term: hispanic Boost: 4
Term: latina Boost: 1
Term: latin Boost: 1
Term: female Boost: 1
Phrase: h f Boost: 1
Phrase: l f Boost: 1
Term: light Boost: 2
Term: blonde Boost: 2
Term: male Boost: 4
Term: thin Boost: 1
Term: skinny Boost: 2
Term: committing Boost: 1
Term: thefts Boost: 1
Term: stealing Boost: 1
Term: shoplifting Boost: 4

The number of records found: 30
The index used: C:\\indexes
The number of documents in the index is: 1641816
Lower : 201701-3553641699 : 8.011668
Lower : 201701-3551757787 : 8.011173
Lower : 201701-3560789631 : 8.010921
Lower : 201701-3581567209 : 8.009311
Higher : 201702-3579142671 : 7.011557
Lower : 201702-3572268329 : 7.0029
Much Lower : 201702-3581550107 : 7.00287
Just Right : 201701-3560086702 : 6.022195
Just Right : 201701-3559044296 : 6.009092
Lower : 201701-3547595233 : 6.007138
```

```
: 201702-3578964639 : 6.007049
: 201702-3579633198 : 6.004963
: 201702-3581106516 : 6.004877
: 201702-3574855935 : 6.004841
: 201702-3581550093 : 6.004827
: 201702-3577525241 : 6.004812
: 201702-3575579555 : 6.004798
: 201702-3576532398 : 6.004774
: 201702-3569674850 : 6.004555
: 201702-3580322486 : 6.00418
: 201701-3560985505 : 6.003694
: 201701-3547314580 : 6.003656
: 201701-3547314704 : 6.003134
: 201701-3552839669 : 5.014605
: 201701-3552839678 : 5.013503
: 201701-3532466453 : 5.013438
: 201701-3562878015 : 5.012779
: 201702-3571219128 : 5.01208
: 201702-3568044996 : 5.011853
: 201701-3574765772 : 5.01183
```

VITA

Larry Snedden is originally from South Florida and currently lives in Jacksonville, Florida. Larry, a veteran of two wars, has served in both the United States Marine Corps and Navy. He has also had the privilege of serving as a Jacksonville Sheriff's Deputy after earning an Associate's Degree in Law Enforcement from Florida State College of Jacksonville. He has a Bachelor's Degree in Information Systems from the University of North Florida (UNF) and expects to receive a Master of Science in Computing and Information Sciences from UNF in the fall of 2017. He has previous experience in the Computing industry as a programmer using C# and Transact SQL. In the fall of 2016 he won a second place award for his data mining submission in the 2016 Clearsense Code-A-Thon. Larry has taught programming at UNF in the School of Computing and currently has worked as a Systems Administrator with twelve years of service.

Larry has many interests outside of Computing. He enjoys charity work, reading, playing guitar, steel fabrication and machining, automobile restoration, and spending time with his family. He has three adult children and one ten year old son. Larry aspires to continue teaching the computing sciences as well as work in the IT industry and contributing to the community.