Query Formulation for Information Retrieval by Intelligence Analysts

Xiangyu Jin[†], *James French*[†], *Jonathan Michel*[‡]
University of Virginia[†], Science Applications International Corporation (SAIC)[‡]
Charlottesville, Virginia
xj3a@virginia.edu, French@virginia.edu, Jonathan.D.Michel@SAIC.com

Keywords: Boolean Query, Information Retrieval, System Evaluation

Abstract

There have been many claims to the effect that advanced information retrieval technologies such as the vector space model (VSM) significantly outperform Boolean search engines in retrieval effectiveness. Most of these claims are anecdotal and rely on arguments such as: queries are difficult to construct or output size is hard to control. In spite of these claims, most operational retrieval systems in use today are Boolean. Some of the popularity derives from the claim that Boolean IR systems are high precision. We compare and contrast Boolean query formulations, including the effect of query expansion, with simply derived VSM models.

1. Introduction

The intelligence analyst process has three major stages: 1) processing information, 2) analyzing information and 3) creation of intelligence products. This paper addresses the issues in the stage of processing information. An important component of processing is the definition of information need and information retrieval. The problem of obtaining a necessary report or study to fulfill an information need is typically accomplished by interaction with an information retrieval system. Requirements on information needs vary widely. There are info needs for a specific critical piece of information, e.g. the Washington D.C. sniper was stopped 10 times at police checkpoints right after the sniper crimes with a valid driver's license and valid out-of-state tags. In contrast there are information needs satisfied by a broad number of documents, e.g. "when is Lincoln's Birthday?" The analyst's problem is complicated by the fact that information is redundant and dynamic. It is redundant due to the fact that many sources will report on the same event or situation. Information is dynamic due to time value where what is reported true yesterday is false today. Processing information is the first step in an effective intelligence process. Effective distillation of the vast breadth of information inherent in an intelligence analyst's problem is critical to the success of the subsequent steps. The processing step is of primary importance. Dissecting the available data, i.e. information retrieval (IR), to characterize and delineate the required information is the gateway to further success in analysis and intelligence production.

The Boolean query is the de Facto methodology used in information retrieval (Frants, et al, 1999). While other approaches exist for retrieving documents, the Boolean approach is the most widely utilized in fielded intelligence systems. The utility of a Boolean approach is often advocated as a high precision technique. The fact that Boolean IR systems are widely used does not mean that they are at the forefront of research. A disconnect exists between the ubiquity of Boolean systems and the ongoing enhancement, study or research in the topic. Gauging by the volume of work in the area of Boolean information systems in academic literature and conferences such as TREC there is low interest in the topic (Sormunen, 2000). Much of the current research in information retrieval focus on non-Boolean approaches such as vector space models, probabilistic models, and other hybrid techniques. Very little is being done looking at advancing Boolean techniques. However, Boolean query is the fundamental technique that is the lynchpin of information retrieval in most operational systems.

This paper looks at the formulation of Boolean queries. We examine both simple approaches and complex approaches with high manual effort for characterizing information need. We examine the intelligence analyst's process in query formulation and in a query refinement. The Boolean queries generated in the study are used as a basis for corresponding VSM queries. The resulting systems are compared.

A difficulty in comparing Boolean Systems is that many of the evaluation criteria are designed for non-Boolean systems. An example is the TREC evaluation criteria. We discuss the shortcomings of TREC measures and approaches. We also suggest a method for comparing performance with other systems and suggest criteria that is appropriate for a Boolean system. Next, we describe the experimentation using several approaches for query formulation: manual and automatic. The results of the experimentation are considered and compared. Finally, the results are discussed and the conclusions are given.

2. Background

2.1. IR Systems

The general task of information retrieval (IR) is searching for information in documents. Here "documents" is a general term, which refers to unstructured records in a database. It can be a text document, an image, a video clip, some web pages, and etc. al. The major difference between an IR search and a traditional RDBMS search lies in the latter one is usually focus on structured data.

Text search perhaps is the most sophisticated area in IR. Its technique usually falls into two categories, statistical approaches and Natural Language Processing (NLP) approaches. The former category usually tokenizes the documents into words, which is the basic element for statistical processing. Variations to this approach extend the role of words to terms . Terms are not restricted to be the words of the documents; examples are n-gram (consecutive string of n characters). A large corpus is usually needed for statistic purpose. The NLP approach employs rules of syntax and semantic level analysis of documents. And NLP is language sensitive. Of course, the boundary is not sharp and these two categories are often interleaved. Statistical approaches dominate operational IR systems. Detailed models includes Boolean, extended Boolean, probabilistic, and vector space.

2.2. Boolean Retrieval Model

The classic Boolean retrieval model is based on logic predicates of terms. Define P(T) as a predicate which asserts that a term T appears within a document. Then we can connect a group of such predicates by AND, OR, NOT relations. The document's relevance is then calculated on the value of these predicates. In practice, the Boolean syntax can be extended. For example, proximity is supported by the predicate. Sometimes, even more complex relation can be defined, such as the sequence of terms or nested proximity where proximate terms must appear within a specified term distance of other terms .

2.3 Query Formulation

The classic Boolean retrieval model is based on logic predicates of terms. Define P(T) as a predicate for whether a term T appears inside the document. Then we can connect a group of such predicates by AND, OR, NOT relations. The document's relevance is then calculated on the value of these predicates. In practice, the Boolean syntax can be extended. For example, proximity is supported by the predicate. Sometimes, even more complex relation can be defined, such as the sequence of terms or terms must appear close enough within a piece of text.

In every query formulation technique there is a human in the loop. From very simple queries to extremely complex queries and there must be a person to define the information need in the form of a query. One of the system performance measures that are often ignored is the level of effort required for query construction. In many cases of the information need, the required query is quite simple. Specifically, simple queries perform well in the case where the information density is high. For example, if the analyst wants to know the score of the Lakers game last night, there are many sources that can provide that information and a simple query will suffice. In other cases, particularly where the information density low, the query must be complex and broad so that relevant data is not missed. Here, iterative automated or human-in-the-loop query formulation techniques are used. The primary technique is query expansion in one of various approaches.

Many query expansion approaches have been attempted. Keyword-type approaches are the predominant type. Four basic types of keyword approaches have been reported. One approach is to use multiple-query searches that are built manually. This human-in-the-loop approach has been used in several research efforts (Carpineto and Romano 2001). Second, a thesaurus method has also been utilized for automatic query expansion (Voorhees 1998). This approach leverages thesaurus for query term expansion. The third methodology is pseudo-relevance (sometimes termed blind-feedback) feedback (Mitra, et al, 1998; Hearst, 1996). Fourth, (French, et al 1998) proposed a relevance feedback approach using a human-in-the-loop. This has been shown to improve query performance in large-scale collections. This approach expands queries using words from the user identified top retrieved documents.

2.4 System Evaluation

Most IR system evaluations have been focused on "continuous measure" best match models. These evaluation systems require the document set to be ordered and ranked with respect to relevance to the query. Traditional measures of performance are used such as interpolated precision versus recall (Harmon, 1994).

There are fundamental problems with inserting a Boolean IR system into this paradigm, i.e. the Cranfield Paradigm, for evaluation (Sormunen, 2000). The basic problem is that the Boolean system creates equivalence classes of returned documents, one class of matched and one class of unmatched documents. Within the classes there is no ranking based on the initial query. Commonly, the matched documents are ranked subsequently; however, this is not a basic feature of the Boolean system. Another issue that equivalence classes create is that the user has no control over the size of the returned set. In the case where there are fewer documents returned than the defined requirement of the evaluation, the Boolean system is penalized.

Within a Boolean IR system the query must be, at least initially, formulated by a human user. This creates the problem of separating the system performance from the user capability. Approaches to address this automatically have been pursued.

The TREC evaluation model, the most often reported system (and the one used in this work), has specific problems for the Boolean IR model. TREC_eval's ground-truth, the qrel files, is not labeled for the entire collection. It is generated from some pooling approach. TREC_eval regard all non-judged (outside the pool) documents as irrelevant. This might make unfair comparison between retrieval systems whose output mostly fall inside and outside the pool.

2.5 Vector Space Model

VSM (Salton and McGill, 1983) is a widely researched but not generally applied (in practice) retrieval model. The basic idea is first create a vector space, whose dimensionality is equal to the number of terms appearing in the corpus. Each document is mapped to a vector, whose component reflects the corresponding term’s weight in that document. This weight can be calculated based on term-frequency in that document (named tf) and the term’s important factor (named idf), which is a global statistic of the corpus. Finally, the query itself is also mapped to a vector and the similarities between query and documents are calculated according to some similarity function. The results are output in a similarity ranked order. There are many variations of vector space model. Different weighting schemes, normalization methods, and similarity functions are proposed within the same framework.

3. Methodology

3.1Testbed Environment

TREC Data

For the experiments reported in this paper we used disks 1 and 2 of the Tipster data used in the TREC-3 evaluation. This consists of approximately one million documents drawn from five sources: AP wire, San Jose Mercury News, Wall Street Journal, Ziff Publishing, and the Federal Register. This data can be considered representative of open source intelligence data.

We used TREC topics 151-200 to form queries for the experiments. These topics have been assessed against the data on disks 1 and 2 and relevance judgments are provided.

Search Engines

We used two search engines in this work: (1) Memex Intelligence Engine,[1] a commercial product; and (2) Lucene,[2] an open source search engine from the Apache Jakarta Project. We used both systems to index the TREC data and both systems were part of the overall query evaluation process as explained below.

Memex is a Boolean IR system while Lucene can be configured as a Boolean system or as a vector space model (VSM) IR system among others.

3.2Query Formulation

The approach used for query formulation is focused on the type of techniques used in the intelligence analyst process. As described above, the TREC data was used. An analyst was given the TREC topics for 50 information needs. This analyst who was experienced in information retrieval created a list of 50 queries. The analyst did not use an iterative approach in creating the queries. He simply expanded the queries by using stemming techniques and straightforward synonym expansion. This set of data is reflective of the quality of queries by an intermediate user. These queries average 38 words in length. This set of queries is labeled Expert-1.

A second analyst was given the same 50 TREC topics and asked to create queries. This analyst has extensive experience in information retrieval. His technique was to start with a relatively simple query then review the top ranked documents. From the documents he selected new query terms that he considered important to the information need. By reissuing the query he refined his search. Each iteration he added both terms that broaden the query and also terms to limit the scope. His technique for limiting irrelevant documents was to review the irrelevant documents in the result set and select terms to add to the query as a negation. This was intended to prevented query drift (Mitra, et al 1998). The query is created by this user are reflective of a highly trained and expert user. This set of data is the highest level of effort required. The queries were complex, averaging 95 words in length. This set of queries is labeled Expert-2.

Next, queries were created by a nominal novice. One of the authors read each of the 50 TREC topics and created a succinct query to address the information need defined in each topic. This set of queries created by the novice simulates the environment where a Boolean IR system is used by an inexperienced user. Often it is the case that an IR system is utilized by a broad range of users. In this case, the queries reflect typical queries created by novices. These queries are typically 2-5 words in length. This set of queries is labeled Novice-1.

[1] http://www.memex.com

[2] http://jakarta.apache.org/lucene

The succinct queries created were used as the basis for an expanded set of queries. These queries were expanded by thesaurus look up with a user in the loop. These queries average 10 words in length. This set of queries is labeled Novice-2.

The final data set is created from the TREC topics themselves. This set of data represents the lowest level of effort in creation. The terms of the topic descriptions were used as queries. The query was created by forming a predicate of the disjunction of all the words in the topic.

3.3 Query Processing

The goal of our experimental methodology is to create a level playing field for two types of query comparison:

1. Boolean query formulations against their "equivalent" VSM queries; and
2. Alternative Boolean query formulations against each other.

This sections describes the steps we took in query processing to achieve that goal.

Vector Queries

Each Boolean query, Q_B, is mapped to a VSM query, Q_V, by removing all Boolean operators and syntax except the prefix search operator[3]. The remaining search terms form a vector query. But it also can be regarded by Lucene as a Boolean query where each term is disjunctive connected. We submit Q_V to Lucene to get a ranklist R_V as the result.

Boolean Queries

Part of the appeal of Boolean queries to experienced users is the degree to which they can exert explicit control over the output of their search. Our expert users were familiar with the search syntax of the Memex system so we felt that they would get the best outcomes by staying in their familiar environment. Initially we intended to map their queries into the Lucene syntax and to conduct all the query evaluation in the Lucene environment. Unfortunately, there was no direct mapping to some features that were considered too important to omit. As a result we executed the Boolean queries on the Memex system to get their unranked results. The evaluation methodology utilizes ranking thus favors the vector model. By ranking the Boolean search output using the same relative ranking that Lucene eliminated this source of vulnerability from our experiments. The ranking used was the same as in the vector query processing for retrieved result R_B.

From that we have, $R_B \subseteq R_V$. The rank of a retrieved document x in the Boolean retrieval list is defined as $s_B(x)$. The rank of the retrieved document in the vector retrieval list is $s_V(x)$, we generate R_B so that

$$\forall x, y[(x, y \in R_B \wedge s_B(x) \leq s_B(y)) \rightarrow s_V(x) \leq s_V(y)]$$

That has the effect on our Boolean retrieved set ranking strategy to produce the same ordering as would have occurred if Lucene had retrieved the same set of documents from a vector query.

Each topic's retrieval result is truncated at a length of 1000 and then is feed into TREC_eval program for evaluation. A complex Boolean queries may result in a smaller result set than a corresponding VSM query. This has the potential to bias some evaluation measures in favor of the VSM, i.e., larger retrieved set. To compensate for this potential we mainly involve the Precision-at as an evaluation metrics rather than the standard Precision-Recall graph.

Novice query:
(regulate | regulation | rating) & (sex | violence) & (movies | video | television)

Expert-1 query:
(regulat* | ban | banning | bans | censor*)(violence | explicit | sex | mature | contain | contains | containing | adult) ((motion picture theat*) | television* | (video*)) (foreign* | domestic* | (united states) | America | oversea* | govern*)(rating | ratings)(newspaper* | magazine* | advertis*)

Expert-2 query:
((ban | banning | bans | censor* | control | controled | controlling | controls | govern | governing | governs | regulat*) & (cassettes | cd | (compact discs):%2 | (compact disks):%2 | game | gaming | internet | (internet games):%2 | (motion picture):%2 | movie | movies | online | (online games):%2 | (T.V.):%4 | tapes | television | TV | (video games):%2 | videos | (web site):%2 | (web sites):%2 | website | websites) & ((adult content):%2 | blood | cursing | cussing | deaths | (drug reference):%2 | (explicit nature):%2 | explicit* | explosi* | goriness | gory | (gun violence):%2 | indecensy | mature | (mature content):%2 | (reference to drugs):%3 | sex | violenc*));%100

Figure 1 - Example queries in the Memex syntax to demonstrate the variation in the query formulation. The default operator is disjunctive. The "%N" syntax is a proximity operator.

The Expected Results

The goal of our experimental methodology is to create a level playing field for two types of query comparison:

1. Boolean query formulations against their "equivalent" VSM queries. In our versions of Boolean query, it is generally believed that these complex syntaxes would help the analyzer to form an effective query, i.e. result in high precision. On the other hand, the vector query

[3] i.e., comput* will match compute, computer, computation, etc.

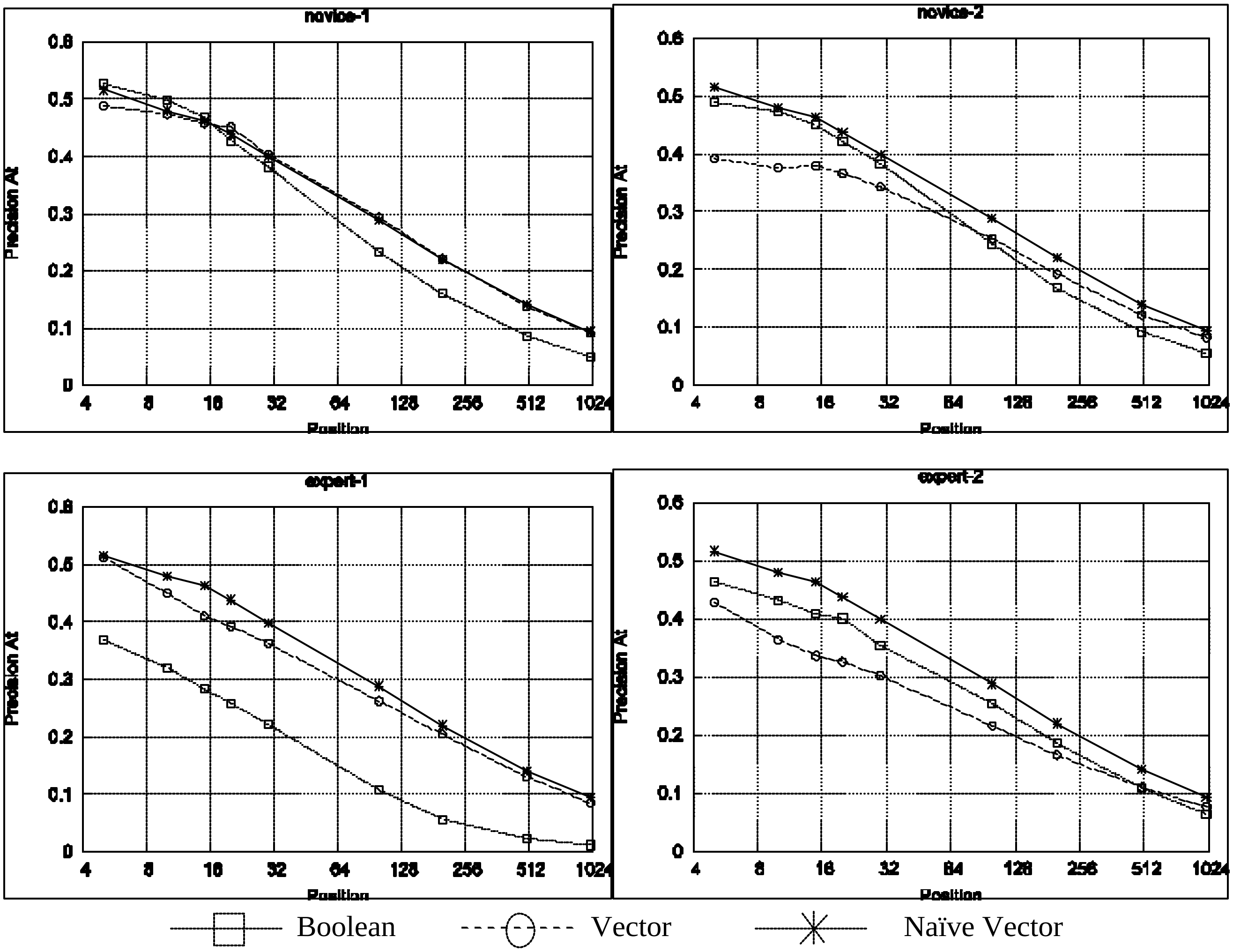

Figure 2 – The top left graph shows the novice-1 results, top right graph displays the novice-2 results, bottom left graph shows the expert-1 and the bottom right shows the expert-2 results. These results are generated from the *trec_eval* program over the 50 TREC-3 queries. The comparison is made using the "Precision-at" measure. This is nominally a valid comparison of disparate IR systems, i.e. Boolean versus VSM. Boolean is the user-created Boolean queries created by the various Novice and Expert users. Vector uses the terms of the individual Boolean queries by forming a disjunctive query and ranking using VSM. The Naïve Vector is a query formed by using the terms of the TREC topic.

is a relax form which would retrieve more relevant documents, which would result in a high recall.

2. Alternative Boolean query formulations against each other. The effort to formulate the query in a descendent order is Expert-2, Expert-1, Novice-2, Novice-1, Naïve-vector. If the work creating complex query formulations is valuable, we would expect their performance ranked in the same order.

4. Experiment Results

Figure 1 shows the results from the novice-1 query set. Note that the left axis shows the precision value and the right axis shows the list position. The list position (Position) is graphed as a log scale. The novice-1 results show a higher value for the precision for the Boolean query at the top of the result list. For example, at five documents deep in the list the average precision is 52% over the 50 queries. The naïve vector, the vector created from the TREC topic directly, is nearly as accurate over all of the list. This reflects higher overall recall in the 1000 document return set.

The novice-2 graph again shows that the naïve vector is highest performing. With the expanded query, the Boolean query is higher performing than the vector query that is based on the same query terms. This shows that in both cases the query term vocabulary is not as complete as with the Naïve query.

The expert query graphs show the benefit of the experienced IR user creating a complex query by iteration. Neither the intermediate nor the experienced user performed as well as the naïve query.

Findings

1. Comparing the Novice-1 versus novice-2 results the decrease in precision seen is expected due to query drift. As the query is broadened the average retrieved set size increases from 312.9 to 398.6. The average relevant retrieved document count increases from 49.1 to 53.4. The slight expansion in recall comes with the cost of reduced precision.

2. Comparing the expert-1 to the expert-2, the expert-2 performance is higher. As in the novice comparison, we have expansion in the retrieved sets. The increase is from an average retrieved set size of 76.6 for expert-1 to 536.8 for expert-2. In the expert-2 case the careful selection of new query terms by the user is a form of relevance feedback. This feedback occurred without access to ground truth. The user inserted terms from documents that he deemed relevant. This helps improve the recall without decreasing the precision. The increase in average relevant documents returned per query was from 11.4 to 63.6.

3. Comparing the novice queries to the expert queries we note that the precision is generally higher for the novice queries. The average query size is smaller and generally more precise.

4. The vector query created from the Boolean Queries described in 3.3, do not consistently improve the results. In some cases they provided substantial improvement, e.g. in the expert-1 query set; however, overall this approach was not consistently better.

5. The naïve query set, formed as a VSM query from the TREC topic is consistently better than either the expert queries or the novice queries. This is the simplest strategy. Rather than reading the topic and deriving a simple or complex Boolean query, the topic itself is used. In this test bed, the user formulation does not add value.

5. Summary and Conclusions

This work set out to study the efficiency and performance of query formulation. Our goal was to review and compare, in a limited study, Boolean query formulation approaches commonly used. We compared the performance of query formulations from three types of users: novice, intermediate and expert. Subsequently, this query data was used to create VSM model queries form the same terms. These results were compared. Finally, a naïve VSM model was created and the results compared with the Boolean approaches. We also considered the problem of evaluation bias in the Cranfield methodology to Boolean approaches. The results were compared using measures that are fair to both VSM models and Boolean models.

The results show that, in this test bed, the effort involved in creating complex Boolean queries does not pay off. Although the manual relevance feedback employed by the expert user increased the performance, the naïve approach produced the best performance. One concern is the phenomenon that the TREC data pooling technique creates. The pooling effect on the evaluation is of unknown magnitude and is a topic of further research.

Acknowledgments

This material is based upon work done while one author (French) was serving at the National Science Foundation. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

C. Carpineto, R. De Mori, G. Romano, and B. Bigi. An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems, 19(1):1--27, 2001.*

Furnas, G. W., Landauer, T. K., Gomez, L. M., and Dumais, S. T. (1987) The vocabulary problem in human-system communication, *Communications of the ACM, Volume 30 Issue 11, November 1987*

Frants, V.I.,; Shapiro, J., Taksa, I. & Voiskunskii, V.G. (1999). Boolean Search: Current State and Perspectives. *Journal of the American Society of Information Science 50(1), 86-95.*

Mitra, M., Singhal, A., and Buckley, C., (1998) Improving Automatic Query Expansion, *In Proceedings of the 21st Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval.*

Ponte, J., Croft, W.B.: A Language Modeling Approach to Information Retrieval. *In: Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press (1998) 275-281*

Vorhees, Ellen M. (1998) Using WordNet for text retrieval. *In Fellbaum C. (ed.) "WordNet: An Electronic Lexical Database", MIT Press*