Comparing Boolean and Probabilistic Information Retrieval Systems Across Queries and Disciplines

Robert M. Losee
Manning Hall, CB# 3360
University of North Carolina
Chapel Hill, NC 27599-3360
U.S.A.

Phone: 919-962-7150
Fax: 919-962-8071
losee@ils.unc.edu

Published in
J. of the American Society for Information Science
48 (2) 1997, 143–156.

June 28, 1998

Abstract

Whether using Boolean queries or ranking documents using document and term weights will result in better retrieval performance has been the subject of considerable discussion among document retrieval system users and researchers. We suggest a method that allows one to analytically compare the two approaches to retrieval and examine their relative merits. The performance of information retrieval systems may be determined either by using experimental simulation, or through the application of analytic techniques that directly estimate the retrieval performance, given values for query and database characteristics. Using these performance predicting techniques, sample performance figures are provided for queries using the Boolean **and** and **or**, as well as for probabilistic systems assuming statistical term independence or term dependence. The variation of performance across sublanguages (used in different academic disciplines) and queries is examined. The performance of models failing to meet statistical and other assumptions is examined.

1 Introduction

Information retrieval is the science and art of locating and obtaining documents based on information needs expressed to a system in a query language. Retrieval systems often order documents in a manner consistent with the assumptions of Boolean logic, by retrieving, for example, documents that have the terms *dogs* and *cats*, and by not retrieving documents without one of these terms. Systems consistent with the probabilistic model of retrieval locate documents based on a query list of terms, such as $\{$*dogs, cats*$\}$, or may accept as input a natural language query, such as *I want information on dogs and cats*. A probabilistic system then ranks documents for retrieval by assigning a numeric value to each document, based on the weights for query terms and the frequencies of term occurrences in documents.

Many lectures and textbooks covering online and CD-ROM searching have taught how to “best” formulate a query, given the searcher’s knowledge of the query and database characteristics. Both practitioners and experimenters have long known that some retrieval techniques work better on certain queries and documents than do other techniques. Realizing this, some researchers, including this author, have combined (with varying degrees of success) the Boolean and term weighting models to produce more general models that are flexible and, ideally, more effective. There have been many theoretical and implementation oriented discussions relating term weighting systems and Boolean-like retrieval, with representative works including [BR84, Cro86, Eva94, EC92, FW90, Lee94, Los94b, LB88, LBY86, Nie89, Rad82, Rad79, RT90, Sal84, Sme84, Tur94]. These models and systems emphasize combining Boolean and probabilistic or vector based systems into a single system. Our discussion below differs in that it trys to keep separate Boolean and term weighting systems that assumes term independence. Given this knowledge, the better search engine for a particular query and database combination can be selected.

As retrieval research has matured, it has become more obvious that different retrieval techniques have different strengths and weaknesses, with some documents most easily retrieved using one technique and other documents being best retrieved using other methods [BCB94, BCCC93, Lee95]. Using these techniques, a system or individual might be able to choose one of several different matching procedures based upon how each procedure will perform.

Our approach to comparing Boolean and term weighting models has become possible because of the development of analytic models of information retrieval performance [Los91, Los95a, Los96]. While most studies of retrieval systems provide experimental results, the ability to precisely describe and explain the operation of Boolean and term weighting systems enables us to calculate the expected performance of either system under a variety of assumptions. In addition, using techniques described below, it will be possible to understand and describe traditional Boolean retrieval as a special case of a term weighting system, enabling us to analytically compare the performance of Boolean and term weighting systems.

This article contains little formal math in the body of the article, with equations provided in the appendix that will be of interest to the more serious student of retrieval.

2 Models of Retrieval and Term Weighting

Several different models of information retrieval systems, including those accepting Boolean expressions as queries, have been popular with commercial vendors and with the information retrieval research community. While the majority of commercial systems have used Boolean query languages, those interested in formal models of retrieval have probably published more on the probabilistic and vector models of retrieval than on Boolean retrieval. The models of probabilistic retrieval provide searchers with a decision rule stating that a document should be retrieved if a calculated value that is based on several parameters is less than a cost based value; if the calculated value exceeds the cost, the document should not be retrieved. These costs are often difficult for patrons to articulate, in part because patrons have little experience valuing or shopping for information, which is so often provided at little or no cost in many societies.

The values calculated in the decision rule provided by the probabilistic model usually require estimating several parameters, denoted here as: p, the probability that a particular term is present in a relevant document; q, the probability that a particular term is present in a non-relevant document; and t, the probability that a particular term is present in a document (ignoring the question of document relevance). A full list of variables is given in Appendix C. Knowledge about parameters is often learned through two processes. The first, the query, provides some information external to the query about what the user expects to find in relevant documents. The information provided by the query is examined in Losee [Los88]. The second way that parameters' values may be estimated is through relevance feedback, information provided by the searcher about what the user finds to be of interest [Boo83, CY82, Los88, Moo93, RTMB86, Sme84, Spi95].

When estimating probabilities for use in the probabilistic model, it is necessary to either assume statistical independence of terms or to formally incorporate some form of statistical dependence between document features [Coo95, CL68, Cro86, LY82, Los94a, Los95a, VR77, YBLS83]. Terms are considered to be independent in most commercial and experimental term weighting systems; when term frequencies are binary, this form of system will be referred to as consistent with the term independence (IND) model. This assumes that, given two terms, one term contains no information about the probability or likelihood that the other term being considered will occur in the same document or in the same relevance class. Different forms of these assumptions have been examined [RSJ76, Rob77], with the most recent discussion provided by Cooper [Coo95]. The term independence assumption is obviously an inaccurate assumption: the term *cats* in a query is obviously more likely to co-occur with terms like *fur* and *dogs* than it is to co-occur with terms such as *tortellini* or *ravioli*. Yet, making the independence assumption allows for the timely and accurate estimation of parameter values and the retrieval of documents.

The probability of two terms co-occurring may be computed from the probabilities of the terms occurring independently as well as a factor representing the degree of dependence between the terms (Equation 2 in the Appendix.) This factor includes c, which represents the expected proportion of documents containing both terms. This variable is similar to a correlation, but it is not the traditional correlation coefficient found in social science statistics. It is the average product of the term frequencies, what might be called the "active component" in the more commonly found Pearson product moment correlation coefficient [Los94a].

Vector retrieval models take a geometric approach to the retrieval problem, with a query and each document being represented as a document vector or arrow moving out from the origin (0,0) point in a space. A document that is very similar to the query will have it's document vector at a small angle to the query vector. The angle between them (computed by a cosine measure) will be relatively small, while documents less similar will have a larger angle between themselves and the query. Each dimension in the "space" may be used to represent the frequencies of a specific term. Assuming term dependence in the vector retrieval model results in the adoption of a somewhat more elaborate model of the term space and the relationship between vectors. Interestingly, in many cases, the formulas developed by those using the probabilistic model are the same as the formulas produced by those using the vector approach [Boo82]. For this reason, we feel comfortable developing our model unifying term weighting systems with Boolean retrieval systems by using the probabilistic model. A similar development using the vector model may be performed, although the interpretation of the processes will be different.

3 Analytic Models of Retrieval Performance

Probabilistic and vector models of retrieval have traditionally been evaluated by simulating retrieval systems using test databases containing sample queries, documents, and relevance judgements. In an analogous manner, one could determine the area of a rectangle through a number of experimental methods, including the simple counting of the number of tiny squares of a given size that one can physically fit in the rectangle. However, many ten years olds, using a formulaic, analytic method, can multiply the height of the rectangle times the width of the rectangle to determine the rectangle's area. The performance of retrieval systems similarly may be determined analytically, with the expected performance being based on a number of factors that directly determine performance, including p, t, and c for the terms included in the query.

The model of retrieval performance used here predicts the *average search length* (ASL), the position of the average relevant document in the ranked list of documents. In situations where retrieval is random, for example, the ASL will be the rank position of the median document in the database. The ASL is used here in lieu of more traditional measures such as precision and recall because it is easy to understand (the ASL is the average number of documents one will need to examine to get to the average relevant document) and because of the ease with which it can be predicted.

The ASL may be computed for the simple case where there is a single term in the query by estimating where the average relevant document would be (probabilistically) in the ranked list of weighted documents, the proportion of the way through the unit space one would need to move to get to the average position for a relevant document. The ASL in the range from 0 to 1 is referred to as the "raw ASL." We begin by estimating the percent of documents without the term and the percent of documents with the term. For each of these categories of documents, we calculate the midpoint of the "percent range" and use it as the representative point for the range. Each of the two points is then multiplied by the percent of relevant documents with the corresponding characteristic. This has the effect of finding the proportion of relevant documents with and without the term, as well as their relative position, in the scaled range from 0 to 1. This value is then multiplied by the number of documents in the database and then $1/2$ is added. A similar procedure may be used for larger numbers of terms.

If there are 100 documents, for example, spreading the relevant documents randomly throughout the database would result in an ASL of 50.5. If there were 5 documents, all with the same term, and the ordering was perfect, the ASL would be 3. Consider a more complicated case where the query is a single term and there are 10 documents, 4 of them relevant, with 3 of the relevant documents and 1 non-relevant document having the single term in the query. Then, $p = .75$ and $t = .4$. The "raw ASL" (scaled in the range from 0 to 1) may be computed by noting the midpoint for the documents with the term is $.2$. This can represent 75% of the relevant documents, while the mid point for the documents with the term, $.7$, can represent the 25% of the documents without the term. Summing $.2 \times .75$ and $.7 \times .25$ produces a raw ASL of $.325$. This, when multiplied by the 10 documents (3.25) can be added to $1/2$, resulting in an ASL of 3.75

These techniques are applied through the rest of the article to a hypothesized database of 100 documents. Random performance for this database will have an ASL of 50.5. As terms become positive discriminators of relevance, that is, p increases beyond t, the performance will move away from this random ASL value. For all the graphs shown here, the variable t was set to $.1$, that is, 10% of the documents have the term in question. All results reported here also use two terms to simplify discussion, with p and t being the same for both terms. This sameness is not required by our model, but accepting the same parameter values for both terms decreases the number of variables present, allowing us to examine and make stronger statements about some other variables of interest (e.g., c.)

The degree of dependence between two terms is captured in part by the average product of the binary term frequencies for the two terms (in all documents). This c value, when used to estimate the unconditioned joint probability (along with the probability t for a single term) is denoted as c_t, while the corresponding c used in estimating the conditional probability that a term pair occurs in a relevant document is denoted as c_p.

The performance results shown in Figure 1 vary parameters p and c_t. These retrieval surfaces show the effect of changes in parameter values, something not easily shown using more traditional experimental methodologies, such as simulation using test databases. Setting $p = .1$ results in the term being a neutral discriminator (resulting in an ASL of 50.5) while a $c_t = .01$ is effectively a Pearson product moment correlation of 0, with a "negative" correlation represented by $c_t < (.01 = t^2)$ and a positive correlation by $c_t > .01$. The surface with the mesh represents performance assuming that the terms are independent and the performance surface without the mesh represents retrieval performance assuming term dependence. For the independence model, the c_p component is computed using Equation 5 in the Appendix so that a Pearson product moment correlation of 0 is modeled between query terms in relevant documents. The dependence based probability is computed with c_p numerically set to that value that results in the lowest ASL, providing an indication of the greatest degree of difference that might be found.

For the retrieval surfaces shown in Figure 1, increasing the c value (c_t) slightly polarizes the documents, increasing the proportion of documents with both terms either present or both absent, resulting in a set of documents that are easier to separate into relevant and non-relevant classes. A stronger effect seen in the Figure is that when p increases, the ASL steadily and more strongly decreases, due to the increase in the discrimination ability of the terms. Performance under dependence in this case is always

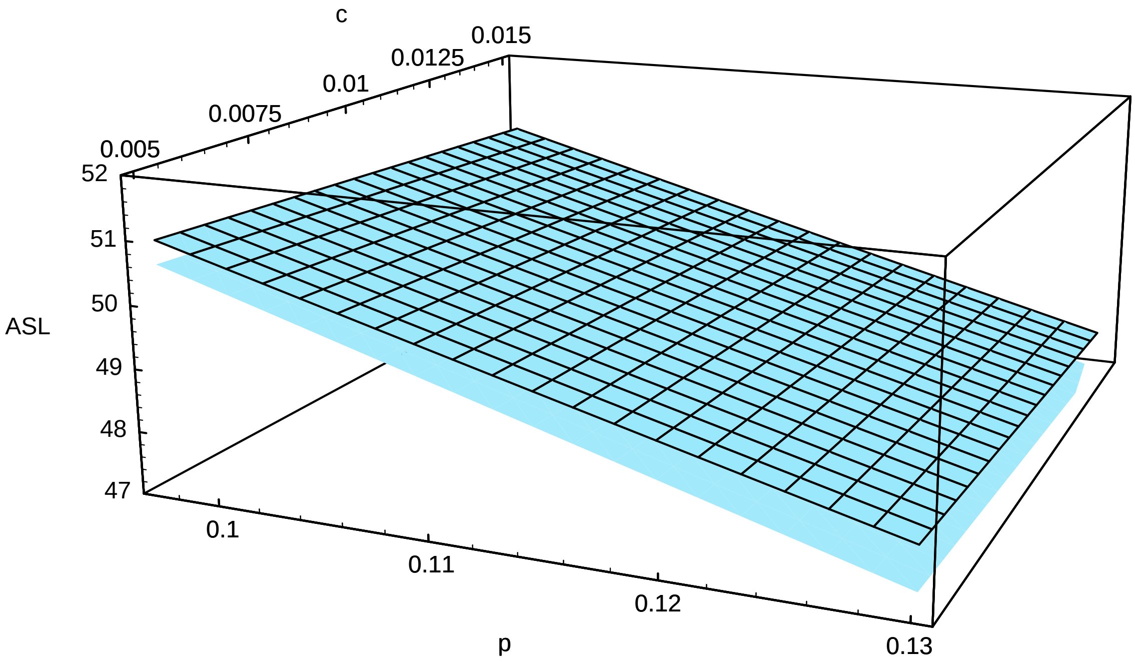

Figure 1: Performance (down is “good” for ASL in these figures) for the term independence model (surface with mesh) and dependence model (surface without mesh.)

superior to or equals that with independence. The worst case performance obtained consistent with dependence assumptions would be that obtained consistent with independence assumptions.

This and some other performance descriptions used here can be said to “mix apples and oranges” in the sense that the c_p values for two different retrieval surfaces are seldom the same for a given c_t value, that is, those points that are directly above one another and have matching c_t and p values often have different c_p values. This is because we are displaying the best performing dependence value consistent with the other parameters used in determining the performance of the independence model.

When comparing term dependence and term independence models, it seems appropriate to assume that the probabilities computed assuming term dependence are accurate, while those assuming term independence are often less accurate. When estimating the ASL, it is necessary when considering each possible document profile (both terms present, both terms absent, etc.) that the probability be computed that the profile is found in relevant documents under the model of interest and using its assumptions. A second probability must be computed that describes the proportion of documents that have this profile; this latter probability is always computed assuming the (accurate) term dependence model.

4 Boolean Operators and Probabilistic Ranking

Retrieval systems based on Boolean logic have long served as the cornerstone of the commercial document retrieval system market and remain very important because of the relative simplicity of the query language and the ease with which it can be understood and implemented. The most common use for a Boolean expression is to state what characteristics must be present in material to be retrieved in a system that retrieves and presents to users bibliographic records or full-text. A second use of Boolean expressions likely to increase in importance over the next decade is in rules incorporated into document and email filtering systems. Such a rule might take the form of a statement, “If the document contains term *foo,* then place document in folder *bar* and display notice of arrival on the screen.”

Boolean expressions typically use three operators: **and**, **or**, and **not**. A search for documents about both dogs and cats might be expressed as *dogs* **and** *cats*. Logical implications, such as *dog* **implies** *mammal,* if something is a dog then it is a mammal, may be expressed without using the implication operator, e.g., using a statement like “It is not the case that something is a *dog* and is not a *mammal*.”

If we are to apply the analytic model of retrieval performance, it becomes necessary to treat Boolean systems as special forms of probabilistic retrieval systems. We suggest a way to do this here, by comparing the ranking provided by individual Boolean operators with the ranking provided by systems consistent with probabilistic models.

4.1 Conjunctive Normal Form

Any Boolean query may be expressed in either of the common normalized forms for Boolean expressions: *Conjunctive Normal Form* (CNF), or *Disjunctive Normal Form* (DNF). CNF represents the conjunction of disjunctions, that is, a series of "**and**ed" components with these components, in turn, consists of the "oring" of individual terms (or the negations of these terms.) Any Boolean expression can be converted into CNF [KK84]. Similarly, a logical expression in DNF is a disjunction of conjunctions, a set of "**or**ed" components, where each component consists of **and**ed terms. Expressions in CNF may be treated as the conjunction of statistically independent components [Boo85, LB88] and may appear more "natural" in some circumstances, although DNF appears to be appropriate in some other circumstances [Cro86].

An expression in DNF such as

$$(dogs\ and\ rabies)\ or\ (cats\ and\ rabies)$$

may be transformed into CNF by regrouping the terms and Boolean operators

$$(dogs\ or\ cats)\ and\ rabies.$$

Another expression,

$$not\ (dogs\ and\ cats),$$

may be transformed into a query in CNF:

$$(\ not\ dogs)\ or\ (\ not\ cats).$$

By converting a Boolean expression of any complexity into one of these normal forms, a ranking of documents using these probabilistic methods may be easily implemented through the simple combination of the methods for the Boolean **or**, **not**, and **and**. We assume below that all our queries have been converted to CNF, thus simplifying the types of operands each of our Boolean operators must accept.

4.2 Boolean "and"

The use of the Boolean **and** may be emulated in a probabilistic retrieval system through the use of joint probabilities and assuming specific term dependencies. Assume a user desires to retrieve documents about *poodles* **and** *allergies*, with those documents not containing both these terms being accorded a secondary place in a ranked list of documents. The same ordering may be desired and obtained using a probabilistic retrieval system if the joint probabilities of *poodles* and *allergies* are estimated in such a way that the documents are ordered so that those with both terms are treated as one set of documents, and those without both terms are retrieved afterward. It is also necessary that a probabilistic system treat documents with either or neither of the terms (but not with both terms) as though they had the same ranking value. The ranking must be:

Term 1	*Term 2*	
1	1	
1	0	
0	1	(these 3 are treated the same.)
0	0	

For a given p, t, and c_t, there is a unique c_p such that the ordering required by the Boolean **and** is obtained. It may be computed using Equation 9, the value for c_p which produces the ranking of documents described above for the Boolean **and**, given the values p, t, and c_t.

Figure 2 shows the level of performance obtained when comparing a query such as X **and**Y with a probabilistic query containing the same two terms, given that the c_p values are computed so that they are consistent with the assumptions described above, i.e. Equation 9. In this figure, the results being compared for a given c_t use two different values for c_p. The c_p value for the Boolean **and** is computed as above, while the c_p for the term dependence model is chosen so that the ASL is minimized.

The relationship between the performance obtained with the Boolean **and** and the term independence model is shown in Figure 3. The values of c_p for both models are computed as described earlier (Equations 5 and 9). Interestingly, the term independence model is sometimes superior to the Boolean model and sometimes inferior. The "break even" point at which both produce the same performance is shown graphically in the figure and may be computed algebraically (Equation 10.)

As one would expect, these results illustrate that the Boolean **and** is not as good as probabilistic retrieval taking advantage of all term dependence information except in a small set of circumstances. At the same time, the **and** results in performance superior to what is obtained assuming term independence when p is not much greater than t and for lower values of c_t. We are comparing the performance expected from two different retrieval models, each of which makes assumptions that are often not met. Figure 3 illustrates which model, with its assumptions, performs better in particular situations.

The Boolean **and**'s performance is often inferior to what we obtain with a system consistent with term independence assumptions because the **and** model treats documents the same whether they have only one term or the other or when they have neither term. However, documents with only one of the terms are more likely to be of interest to the user than a document with neither term. The probabilistic model can rank those documents with only a single term in common with the query between those documents with both terms and those with neither term, resulting in performance superior to that obtained with the Boolean model.

The term independence model is sometimes inferior due to the performance obtained when the c value drops. When there is a negative correlation between terms (i.e.,

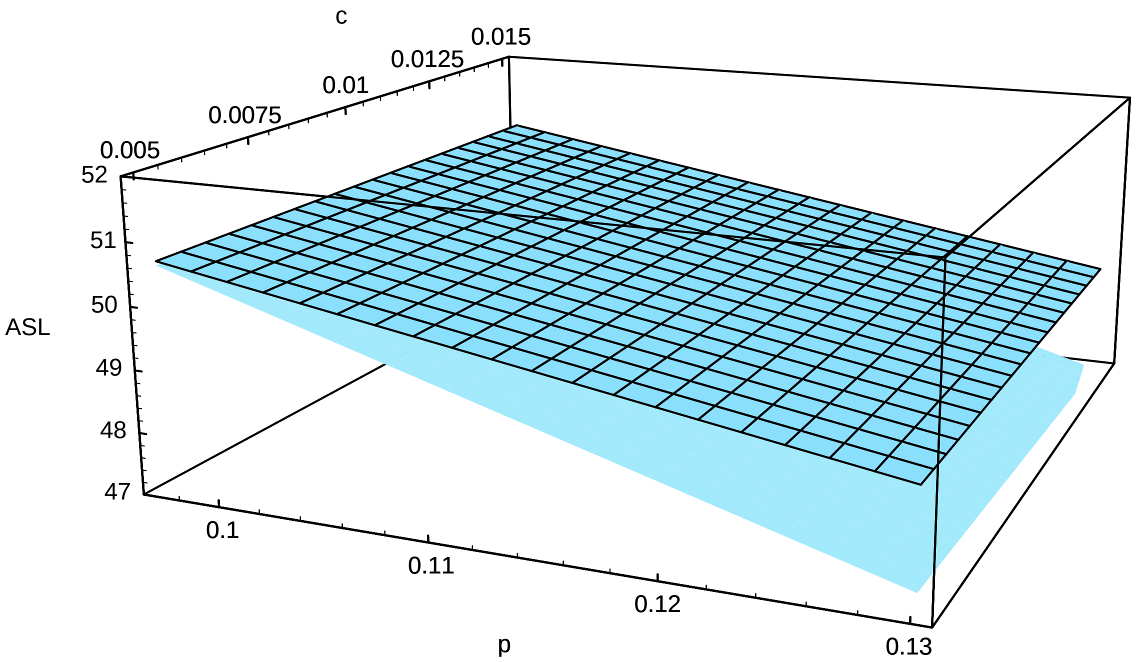

Figure 2: The meshed surface represents the performance obtained with the **and** model and the unmeshed surface represents the (superior) performance obtained with term dependence. The value p represents the probability for both terms that they occur in a relevant document, assuming dependence, while c is a form of correlation measure, the average product of the terms in all documents $(c_t.)$

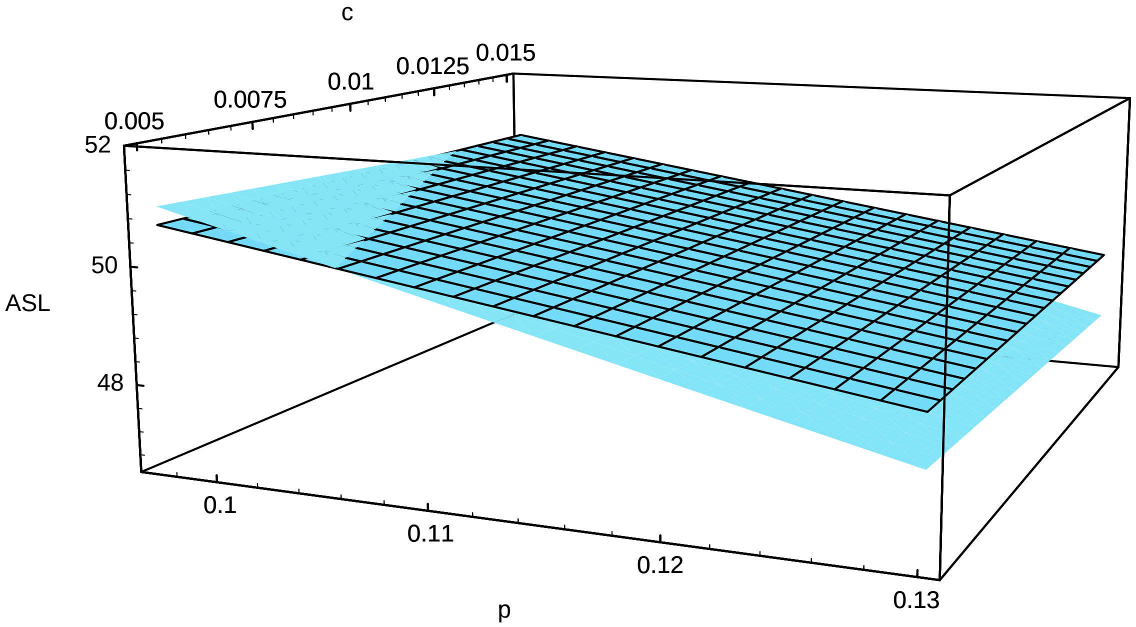

Figure 3: The performance of the Boolean **and** is displayed as a meshed surface, with the term independence model displayed as a plain surface.

$c_t < t^2$), fewer documents that have one term will have the other, making it harder to discriminate between the relevant and non-relevant documents.

4.3 Boolean “not”

The Boolean **not** may be emulated through the reversal of the probabilistic ranking for the term in question. For example, processing the Boolean query *quilting* may be emulated by retrieving documents based on the probabilistic method applied to the term *quilting.* The Boolean query **not** *quilting* may be emulated as the reverse of the ordering obtained with the query *quilting*. This is the same as ordering the documents by the probability that a document doesn’t have the term. When queries are in conjunctive normal form, as we are assuming they are, they will be at most one term as the operand for each **not** operator, and thus we need only address single terms as operands for **not**. The performance of the term independence and Boolean models will be the same for simple **not** operations because they each have a single term or concept used in document ordering.

4.4 Boolean “or”

The use of the Boolean **or** also may be emulated by a probabilistic information retrieval system (as with the Boolean **and**) through the use of joint probabilities and by assuming specific term dependencies. A query consisting of two terms connected with the Boolean **or** operator retrieves documents having either one or both of the terms. The same ordering may be obtained using a probabilistic retrieval system if the joint probabilities of the two terms are selected so that the documents are ordered so that documents with either (or both) of the terms are treated as one set of documents, and those documents with neither term constitute a second set to be retrieved afterward. The ranking must then be:

Term 1	*Term 2*	
1	1	
1	0	these 3 are treated the same
0	1	
0	0	

As with the Boolean **and**, there is a unique c_p for a given set of parameters. This c_p may be computed using Equation 13, which determines the necessary value for c_p which produces the ranking of documents described above for the Boolean **or**.

Figure 4 shows the level of performance obtained when comparing a query such as X **or**Y with a probabilistic query containing the same two terms and that the c_p values are computed so that they are consistent with the assumptions described above, i.e. Equation 13. Two different values for c_p are used in producing this figure. The c_p value

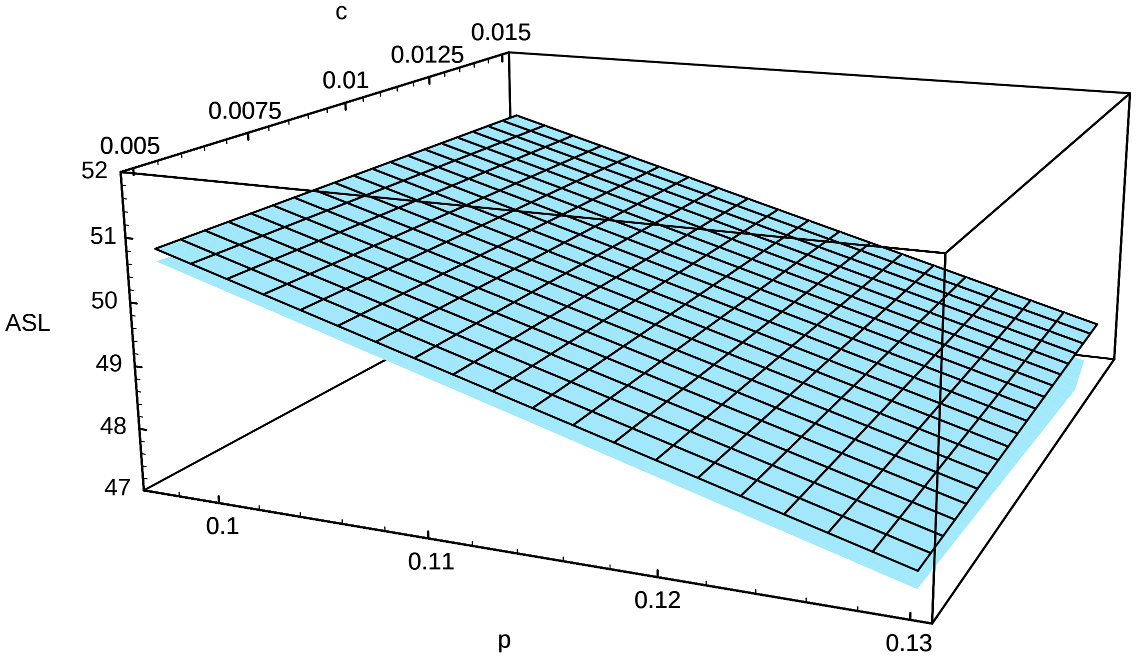

Figure 4: The meshed surface represents the performance obtained with a Boolean **or** and the unmeshed surface represents the (superior) performance obtained with term dependence.

for the Boolean **or** is computed consistent with Equation 13, while the c_p for the term dependence model is again chosen so that the ASL is minimized.

The relationship between the performance obtained with the Boolean **or** and the term independence model is shown in Figure 5. Performance using the term independence model is sometimes superior to the Boolean model and sometimes inferior. The "break even" point at which both produce the same performance is shown graphically in the figure and may be computed algebraically (Equation 14.)

As one would expect, these results illustrate that the Boolean **or** is not as good as probabilistic retrieval taking advantage of all term dependence information. At the same time, the **or** sometimes results in performance superior to what is obtained assuming term independence.

The Boolean **or**'s performance is often inferior to what we obtain with a system consistent with term independence assumptions because the **or** model treats documents the same whether they have only one term or both terms. The model can be said to "throw away" the obvious information that a document with two query terms is more likely to be of interest to the searcher. The probabilistic models are not limited by this constraint and can take advantage of this information.

5 Retrieval Variations across Disciplines and Queries

Academic disciplines vary in how their scholars express ideas, varying from highly quantitative expressions in mathematics to complex nominal expressions found in biological literature to the unique and precise use of common terms such as "truth," "knowledge," and "beauty" by philosophers. The differences between the languages used in academic disciplines (sublanguages) have been the object of considerable study [Bec87, Bon84, Haa95, LH95, Sag81, Tib92]. Some of the differences between disciplines may be viewed on a spectrum from the hard to the soft sciences [LH95]. The disciplines may also be viewed in terms of those fields that create and donate concepts and terms as against those disciplines that are net borrowers [Los95b].

Using the analytic techniques described above, we have attempted to examine the effectiveness of different retrieval procedures on different disciplines. Using a database developed by Stephanie Haas [Haa95, LH95], term frequencies and the relationships between terms are computed and examined in the light of the requirements for each of several different retrieval procedures. The database consists of approximately two hundred titles and abstracts from each of eight disciplines: biology, economics, electrical engineering, history, mathematics, physics, psychology, sociology. For the research described in Losee and Haas [LH95], lists of terms were randomly extracted from each of the eight databases and their status as sublanguage terms studied. Terms on these lists are categorized (for purposes here) as sublanguage terms when they match in part a dictionary entry from a subject dictionary and are labeled as general language terms if they do not match in part an entry in the subject dictionary. Sublanguage terms are

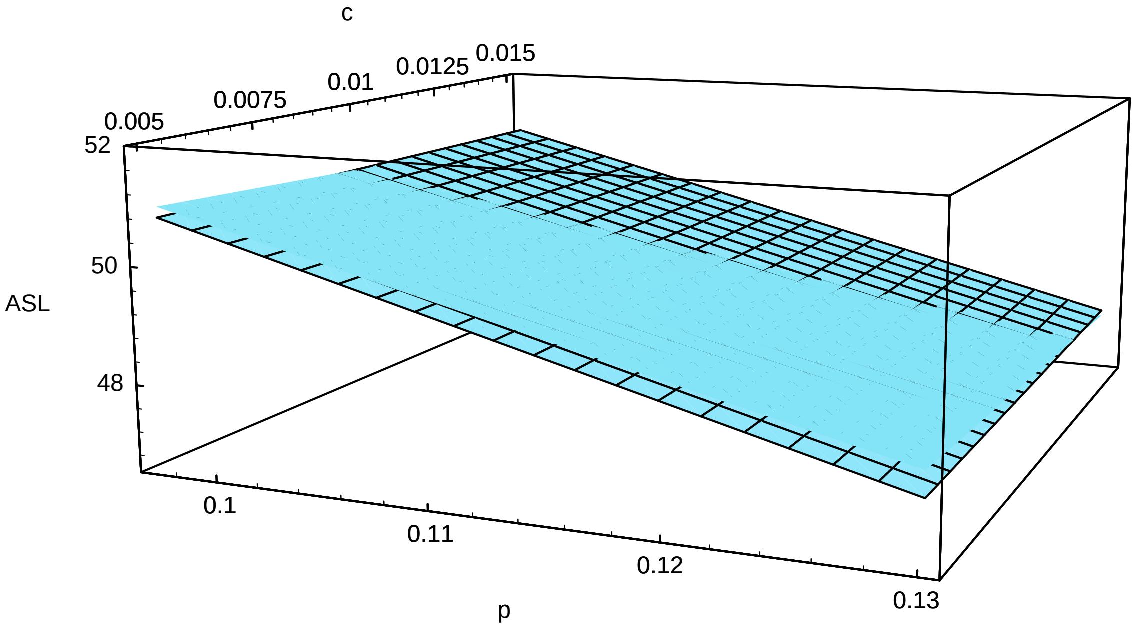

Figure 5: The meshed surface represents retrieval performance with the Boolean **or** and the unmeshed surface represents the probabilistic IND model’s performance.

Discipline	p	c_p	t	c_t	*Disc.*
See printed article for tabular data					

Table 1: Parameter values for different disciplines. *Disc.* represents the traditional IND term weight (discrimination value) computed using the p and t values.

considered here as a pool of subject related terms likely to be found in a query, and these terms are used below as sample query terms from which all possible pairs of sublanguage terms are derived. These sublanguage term pairs may be used in estimating p and c_p parameters, while a random set of tens of thousands of term pairs were used in estimating t and c_t parameters. For this study, terms were stemmed using the Porter [Por80] stemming algorithm.

Table 1 presents the average p and t values for the eight disciplines, along with the correlation-related parameters c_p and c_t. The value on the right hand side of the table is the discrimination value of the "average" term, which is at its maximum for mathematics and sociology, with its low point for biology. The highest values for p are similarly found in mathematics and sociology. The variable c_t is highest for the fields of biology and economics. An examination of a ranked list of the c_t values for term pairs from all disciplines found that of the top twenty five term pairs, fifteen were from biology and eight were from economics, with only two coming from other disciplines (psychology and electrical engineering.) Because of the small sample sizes for the "sublanguage" terms, we don't make strong claims about specific disciplines or about the characteristics of hard or soft sciences. The results here are meant to provide examples of what may be anticipated from a diverse set of disciplines.

Different retrieval models may be evaluated after making explicit the assumptions of each particular retrieval model. If the assumptions are met, such as those made by Equations 9 or 13 for the Boolean **and** and **or**, respectively, then Boolean models can be described probabilistically. Performance is estimated by computing the ASL for a particular model and comparing these ASL values for each method, given a set of parameters. When computing the ASL (Equation 1), two types of probabilities are used: those describing the actual data (unconditional probabilities) and probabilities conditioned upon relevance. The conditional probabilities may reflect the probabilities as seen by the model being evaluated, while the unconditional probabilities may be understood as describing the actual distribution of data in all documents, as (precisely) predicted by the term dependence model. In cases where we know the degree of term dependence across all documents, i.e., c_t, we may calculate the performance of a model assuming the c_p value suggested by the model. This performance may be compared with the performance obtained with the actual c_p value, with term dependence.

Using the assumptions above, the type of retrieval model (e.g., **and**, **or**, or IND) only effects the retrieval performance model through the parameter estimates for c_p. For

Discipline	IND	**and**	**or**
Soc	6.56	92.51	0.93
Math	8.51	87.89	3.60
Econ	10.48	85.44	4.08
Bio	10.55	84.92	4.52
EE	11.18	85.34	3.48
Phys	12.84	83.78	3.39
Psych	15.05	81.58	3.37
Hist	17.78	78.19	4.03

Table 2: The percent of randomly selected sublanguage term pairs for different disciplines in which the retrieval model indicated is best. This assumes the parameters of the model are consistent with the retrieval model (i.e., Equations 5, 9, or 13).

Use Method	**(Assuming)**	**When**
Boolean **and**	Equation 9	$\Delta_{IND}^{and} \leq 0$, Equation 11 & $\Delta_{and}^{or} \geq 0$, Equation 16
Boolean **or**	Equation 13	$\Delta_{IND}^{or} \leq 0$, Equation 15 & $\Delta_{and}^{or} \leq 0$, Equation 16
Term Weighting (IND)	Equation 5	$\Delta_{IND}^{or} \geq 0$, Equation 15 & $\Delta_{IND}^{and} \geq 0$, Equation 11

Figure 6:

each set of values for c_t, t and p, there is a c_p value that provides *optimal performance.* Using another value for c_p, such as one likely to be assigned by one of the various retrieval models, may provide performance significantly below that which could be obtained with optimal models. Examining the difference between retrieval performance with the actual parameter c_p and the parameter as estimated by the model can provide an indicator of the performance level using a model's parameter estimate of c_p.

Table 2 shows the percent of randomly selected sublanguage term pairs that are best retrieved with each of the three retrieval models, given that the assumptions about c_p for each of the models (e.g. Equations 9 or 13) hold. The variable p is the discipline average shown in Table 1 and the values c_t and t are taken from the individual term pairs. The decision rules used are provided in Figure 6. The value $\Delta_{M_2}^{M_1}$ denotes the difference in performance between retrieval model 1 and model 2. When this difference is positive, the first model has a higher (worse) performance than the second one, while when it is negative, performance with the second model exceeds the first, measured in terms of

ASL, and the first model is the better performer. These are the rules used to produce the results shown in Table 2.

As an example of using these decision rules, let us assume that both terms have $t = .1$. One may then calculate the Δ values (and the better model, indicated in parentheses) for these two cases:

 Δ	$p = t + .01$ $c_t = t^2$	$p = t + .05$ $c_t = t^2 + .01$
Δ_{IND}^{and}	$+0.0038$ (IND)	$+0.0208$ (IND)
Δ_{IND}^{or}	-0.0004 (**or**)	$+0.0003$ (IND)
Δ_{and}^{or}	$+0.0040$ (**or**)	$+0.0178$ (**or**)

When the degree of dependence between terms is increased (moving from the left column to the right), along with an increase in the discrimination value of the terms, it becomes clear that the retrieval performance using the Boolean **and** drops compared to that obtained with the term independence model or the Boolean **or**. Here the Boolean **or** is superior to the **and**.

Using a different approach with the eight databases described above and with the parameter values shown in Table 1, we find that when all the assumptions of each Boolean model are met, and ignoring Equations 5, 9, and 13, the results shown in Table 3 are obtained. Here the actual parameters for the data are used and the document groupings imposed by the different models are met here. Instead of using the estimates of c_p for the Boolean models, provided by Equations 5, 9, and 13, the average experimentally determined value for c_p for each discipline is used. Using the discipline averages for c_p instead of the values for each individual pair may have a significant influence on the performance results obtained.

When retrieving using the Boolean **or**, for example, the values for parameters p, t, c_p and c_t are obtained from the data, unlike the earlier techniques used in producing Table 2 that generated c_p so that it fit certain assumptions. Documents with **or** are ranked so that documents with either (or both) of the query terms are treated as being of the same rank, while those documents with neither term are ranked after these first documents. This is different than the method used in producing Table 2, which used Equations 9 and 13 to estimate c_p for the Boolean models, and all different document profiles are treated as unique. When the values for parameters meet the requirements of Equations 9 or 13, ranking obtained from a dependence based probabilistic system will be the same as with a Boolean system because of the parameter values. The data in Table 3 are not produced by assuming that Equations 9 or 13 hold but, instead, the iteration (summation) through the documents is modified so that the proper groupings are obtained (required for the probabilistic model to provide ranking identical to that provided by a system using the Boolean model.) This modification computes the ASL (Equation 1), collapsing certain sets of terms together for specific Boolean models.

Discipline	IND	**and**	**or**
fboxSee printed article for tabular data			

Table 3: The percent of randomly selected sublanguage term pairs for different disciplines in which the model indicated is the most appropriate retrieval model. All the requirements of each model are met.

6 Summary and Conclusions

Several decision making techniques have been suggested for the practitioner who has the option of choosing which search language or retrieval model to use for a given database, in situations such as when a database is searchable through more than one vendor's search engine. The individual searcher may select the search mechanism expected to perform best, or an automated system may make the decision for the user or assist the user.

The most important recommendation we can make to searchers is that systems consistent with term dependence models should be used, whenever possible. While performance is improved by taking advantage of the statistical dependence between term frequencies, it may be computationally too expensive in many instances for system designers and implementors to incorporate term dependence as an option [Los94a, LBY86]

Searchers need to carefully consider the assumptions that are made by the Boolean models. For example, in the case of **and**, all documents without both the terms being **and**ed are assumed to have the same chance of being of interest to the user, which is usually a bad assumption to make. Similarly, the Boolean **or** treats all documents with either of the **or**ed terms as being identical in importance, which is also unrealistic. In general, documents with only one of a pair of terms are somewhat less valuable than a document with both terms and are correspondingly more valuable than a document with neither of the terms. As was seen above, the documents are only equal in value, having different sets of characteristics, when specific parameter values exist, which is seldom the case.

Table 3 suggests that the Boolean **and** produces superior performance to the Boolean **or** in many but not in all cases. This superiority, if it can be shown to hold in other databases and with more realistic query sets, would suggest that systems that supply their own operators should usually supply the Boolean **and** rather than the **or**. Expressions in CNF may be more natural for end users, who might be further encouraged to use **and**. This is consistent with the historical emphasis on the conjunctive normal form and the frequent use of the Boolean **and** by searchers [Gup87, MC81].

The relationship between the Boolean **and**, the **or**, and the IND model may be understood in terms of the nature of the correlations between terms. For example, the Boolean **and** requires that documents with either term or neither term be treated identically by the ranking process. The rankings of disciplines with regards to the different models in Table 3 seems to be related to the ordering obtained with p or c_p in Table 1.

The ranking in Table 3 obtained with the Boolean **or** is only equivalent to the ranking obtained with the Boolean **and** when there is a perfect positive correlation between the two terms. When this occurs, the presence or absence of one term will dictate the presence or absence of the other. There will not be any documents with only one of the terms. This provides the same ranking under Boolean **and** and **or** by effectively *eliminating* all documents except those with both or with neither terms: both these types of documents have the same relative rank position under ordering with **and** and with **or**.

Boolean retrieval systems can be studied analytically as special cases of probabilistic retrieval systems. Statements composed of a single Boolean operator can be modeled probabilistically and compound Boolean statements can be treated as nested simpler Boolean expressions placed into CNF. Different forms of queries may be compared and optimality examined [Los94b].

Due to the differences in sublanguages used in academic fields, we may expect some differences in the types of queries that are best suited to each discipline. The data discussed show that disciplines do vary in the degree to which different retrieval models should be used. Due to problems associated with the small number of abstracts and the small number of sublanguage terms used in the sample, it would be unwise to make any generalizations about those search techniques most appropriate for specific disciplines.

In summary, situations in which using the Boolean **and** or the **or** produce the best results can be analytically determined through the use of appropriate formulae using the values of individual and joint term probabilities and frequencies. One method of determining the best model to use requires that we take advantage of all the information available about the database and the statistical evidence about term relationships for those terms in the query. Using this information, and given the option of the same database being available on multiple systems, one with a Boolean search mechanisms and one with probabilistic search capabilities, the system with the best expected retrieval performance may be selected. While the term dependence model will almost always be superior, it is seldom the case that this model is available on existing systems. Instead, one must often choose between systems consistent with simple term independence models and systems consistent with Boolean assumptions. Using the methods described here, it becomes possible to estimate which method will be superior, given actual or estimated parameter values.

A Appendix – Computing Retrieval Performance

The Average Search Length (ASL) may be computed in different ways, depending on the different retrieval assumptions with which it is desired that the ASL be consistent

[Los95a, Los96]. For our two term model, both terms are assumed to have the same probability of occurrence (t) in all documents and the same probability of occurrence (p) in relevant documents. We may compute the ASL thus

$$ASL = N \left[\sum_{i=11}^{00} \Pr(d_i|rel) \left(\sum_{j=11}^{Pred(i)} \Pr(d_j) + \frac{\Pr(d_i)}{2} \right) \right] + .5, \tag{1}$$

where $Pred(i)$ is the predecessor of profile i, the iteration proceeds in the order $11, 10, 01, 00$, and N is the number of documents in the database. This assumes that both terms are positive discriminators, making the order progress from documents with the terms to documents without the terms.

The probabilities (in Equation 1) that a document occurs in either relevant ($\Pr(d_i|rel)$) or in all documents ($\Pr(d_i)$) may be computed using the Bahadur Lazarsfeld Expansion (BLE), which may be used to estimate (or compute exactly) probabilities of term pairs, triples, etc. Described for the general case in [YBLS83, Los94a], we consider here the BLE for the two term case, that is, when we wish to compute $\Pr(x, y)$. For this two term case, the BLE takes the following form:

$$p^x(1-p)^{1-x}q^y(1-q)^{1-y} \left[1 + \frac{E\left((x-p)(y-q))(x-p)(y-q)\right)}{pq(1-p)(1-q)} \right] \tag{2}$$

where p and q are the probabilities that the first and second terms have term frequencies x and y of 1, and where $c = E((a)(b))$ is the expected value for the product a and b.

If the terms are independent, the probability that two terms occur with frequencies x and y, respectively, is

$$p^x(1-p)^{1-x}q^y(1-q)^{1-y}. \tag{3}$$

Adding the assumption that the terms are dependent requires that the following component be added to (3):

$$p^x(1-p)^{1-x}q^y(1-q)^{1-y}(x-p)(y-q) \quad \frac{E\left((x-p)(y-q))\right)}{pq(1-p)(1-q)} \tag{4}$$

Note that the fraction on the right is a constant for all document profiles.

When computing the probabilities of term pairs occurring consistent with the term independence model, it is necessary to estimate or compute $\Pr(d_i|rel)$, using c_p as the c component, the numerator of the fraction on the right hand side of Equation 4, and to compute $\Pr(d_i)$ with the c component represented by c_t. Given knowledge of c_t, we may compute c_p consistent with the independence assumptions

$$c_p^{\mathbf{IND}} = p^2 \tag{5}$$

The true values of p, t, and c_t are used when c_p is estimated. Using this estimate of

c_p, the same ranking is obtained with the term dependence model as would be obtained using the traditional term independence model.

B Boolean Expressions & Probabilistic Ranking

The ranking obtained with Boolean expressions may be emulated probabilistically if a document weighting system is used that is consistent with probabilistic models of retrieval and incorporates feature dependencies. Using the analytic model of retrieval performance, we can learn those situations where a Boolean query or Boolean system would be superior to using a probabilistic system and when the probabilistic system would be superior to the Boolean system.

We first consider the simplest form of logical expression, a single term such as x, with no operators. The ranking of documents using the Boolean expression x retrieves first documents with x, followed by documents without term x. We denote this Boolean ranking as $B(x)$ This same ranking is obtained by a probabilistic retrieval model with a single term. The probabilistic ranking based on probabilistic weight $P(x)$ is denoted as $R\Big(P(x)\Big)$. For our purposes, we will denote this equivalent ranking by the $\Longleftrightarrow$ symbol, thus

$$B(x) \Longleftrightarrow R\Big(P(x)\Big). \tag{6}$$

B.1 Negation

The simplest logical operator is negation. A Boolean query such as **not** *x* (denoted as $\neg x$) simply requires that the ranking be reversed from that obtained with x. The probabilistic ranking is simply the inverse ranking from that obtained with $P(x)$, that is, the ranking obtained with $P(\neg x) = \overline{P}(x)$. Ranking with the *rule of logical negation* is thus

$$B(\neg x) \Longleftrightarrow R\Big(\overline{P}(x)\Big). \tag{7}$$

In some circumstances, an odds formulation may be more mathematically tractable or useful. One might then compute

$$B(\neg x) \overset{odds}{\Longleftrightarrow} R\Big(\left(1 - P(x)\right)/P(x)\Big).$$

Similarly, the odds formulation of equation 6 is

$$B(x) \overset{odds}{\Longleftrightarrow} R\Big(\left(P(x)\right)/\left(1 - P(x)\right)\Big).$$

The use of such odds based formulae address some of the concerns that have been raised about term independence models [Coo95, RSJ76]. These concerns are not an issue in models that assume and fully compute term dependence.

B.2 Conjunction

The retrieval characteristics of a Boolean system presented with a Boolean query consisting of the conjunction of two terms may be emulated by the ranking provided by probabilistic retrieval if one accepts the *rule of ranking conjoined features*:

$$B(x \wedge y) \Longleftrightarrow R\Big(P\,(x, y)\Big). \tag{8}$$

In this expression, $P\,(x, y) = \Pr(rel|x, y)$. The odds form of Equation (8) is

$$B(x \wedge y) \Longleftrightarrow R\left(\frac{P\,(x, y)}{(1 - P\,(x, y))}\right).$$

The extension of these rankings beyond two terms is simple.

The ranking provided by $B(x \wedge y)$ may be represented in a probabilistic ranking by using the joint probability $P\,(x, y)$, that is, the probability that x and y will occur together. While this approach is intuitively appealing, it brings up a problem that occurs with multiterm systems. Emulating the ranking of the Boolean system within a probabilistic system requires that there only be two types of documents, those with both x and y, and those with *either* one *or* the other *or* neither. It thus becomes necessary to find a probabilistic formulation consistent such that $R\Big(P\,(x = 1, y = 0)\Big) = R\Big(P\,(x = 0, y = 1)\Big) = R\Big(P\,(x = 0, y = 0)\Big) \neq R\Big(P\,(x = 1, y = 1)\Big)$. This is obtained if we assume that $P\,(x = 1, y = 0) = P\,(x = 0, y = 1) = P\,(x = 0, y = 0) \neq P\,(x = 1, y = 1)$. A simple solution to this begins with equating the probabilities of terms $x = 1$ and $y = 1$, making $P\,(x = 1, y = 0) = P\,(x = 0, y = 1)$. We then determine a degree of dependence between the two terms such that $P\,(x = 1, y = 0) = P\,(x = 0, y = 0) = P\,(x = 0, y = 1)$. Obviously, this can only be true if the second order dependence can "counter" the change in one of the other parameters.

When computing the term dependence for term pairs in the Boolean models described in the body of the paper, it becomes necessary to compute $\Pr(d_i|rel)$ using c_p as the c component, described in Appendix A, and to compute $\Pr(d_i)$, where the c component is represented by c_t. Beginning with knowledge of c_t in the Boolean **and**, we may compute c_p consistent with the above assumptions as:

$$c_p^{\mathbf{and}} = 1 + \frac{-1 + c_t + p - c_t p}{1 - t}. \tag{9}$$

When one ranks documents assuming term dependence, the true values of $p, t,$ and c_t are used, and c_p is estimated as in Equation 9, giving the same ranking as would be obtained with a Boolean query using the **and** operator.

Assuming this method for computing c_p, one can compute the break even point between the Boolean **and** and the term independence model assuming Equation 5, that is, where the two models produce the same retrieval ordering. We denote this as

$$c_{\Delta^{and}_{IND}=0} = \frac{-p + p^2 + t + 2pt - 3p^2t - 2t^2 + 2p^2t^2}{1 - p - 2t + 2pt}. \tag{10}$$

More generally, the c_t value needed when a difference Δ^{and}_{IND} in the "raw ASL" on the unit interval from 0 to 1 (that is, the ASL before it is multiplied by the number of documents and $.5$ added) is desired between the raw ASL obtained when using the Boolean **and** and the raw ASL obtained when using the term independence model, is computed as

$$c_{\Delta^{and}_{IND}} = \frac{-p + p^2 + t + 2pt - 3p^2t - 2t^2 + 2p^2t^2 + 2\Delta^{and}_{IND} - 2t\Delta^{and}_{IND}}{1 - p - 2t + 2pt}. \tag{11}$$

Note that other kinds of break-even points may be computed consistent with different assumptions.

B.3 Disjunction

When we say that x is true or y is true, we can also say that it is not the case that x is false and that y is false at the same time, that is, both can't be false if x is true or y is true. We can thus say that $(x \vee y) = \neg(\neg x \wedge \neg y)$. The *rule of ranking disjoined features* is:

$$B(x \vee y) \Longleftrightarrow R\left(\overline{P}\,(x = 0, y = 0)\right) \tag{12}$$

Using methods similar to those used with the Boolean **and**, we may compute that when the Boolean **or** operator connects the two terms, the c_p value must be

$$c_p^{\mathbf{or}} = c_t p/t \tag{13}$$

Assuming that c_p is this value (for the next three equations), the break even level of performance between a query using the Boolean **or** and the term independence model is

$$c_{\Delta^{or}_{IND}=0} = pt \tag{14}$$

Given Δ^{or}_{IND}, the difference in the raw ASL performance between the Boolean **or** and the term independence model, we may compute c_t as

$$c_{\Delta^{or}_{IND}} = \frac{t(p^2 - 2p^2t + 2\Delta^{or}_{IND})}{p - 2pt}. \tag{15}$$

Using a similar technique, the c_t value may be computed, given the (Δ^{or}_{and}) by which the raw ASL for the Boolean **or** exceeds the raw ASL for the Boolean **and**, and given that c_p is computed consistent with the assumptions of Equations 9 or 13 in the case of the Boolean **and** or **or**, respectively. We compute c_t thus:

$$c_{\Delta_{and}^{or}} = \frac{t(-p + t + 2pt - 2t^2 - 2\Delta_{and}^{or} + 2t\Delta_{and}^{or})}{-p + t + 2pt - 2t^2} \tag{16}$$

Using these equations to determine Δ values allows one to analytically determine the degree of performance increase or decrease that would be obtained in circumstances consistent with the assumptions described earlier by moving from one model to another.

C List of Variables

rel	Of use to the user.
p	Probability a term occurs in a relevant document.
t	Probability a term occurs in a document.
c	Expected frequency of the product of two terms.
c_t	c for the set of all documents.
c_p	c for the set of relevant documents.
$\Delta_{M_2}^{M_1}$	Difference in “raw ASL” between retrieval models, $M_1 - M_2$.
$B(x)$	The ordered set of documents produced by Boolean query x.
$P\left(x\right)$	The probabilistic value attached to a document given query x.
$R\left(P\left(x\right)\right)$	The ranking based on the set of probabilistic values attached to documents given query x.
$M_1 \Longleftrightarrow M_2$	Method M_1 produces the same document ranking as method M_2.

References

[BCB94] B. T. Bartell, G. W. Cottrell, and R. K. Belew. Automatic combination of multiple ranked retrieval systems. In *ACM Annual Conference on Research and Development in Information Retrieval*, pages 173–181, New York, 1994. ACM Press.

[BCCC93] N. J. Belkin, C. Cool, W. B. Croft, and J. P. Callan. The effect of multiple query representations on information retrieval perforamnce. In *ACM Annual Conference on Research and Development in Information Retrieval*, pages 339–346, New York, 1993. ACM Press.

[Bec87] Tony Becher. Disciplinary discourse. *Studies in Higher Education*, 12(3):261–274, 1987.

[Bon84] Susan Bonzi. Terminological consistency in abstract and concrete disciplines. *Journal of Documentation*, 40(4):247–263, 1984.

[Boo82] Abraham Bookstein. Explanation and generalization of vector models in information retrieval. In *Research and Development in Information Retrieval: Proceedings of the 5th International Conference on Information Retrieval*, pages 118–132, Berlin, 1982. Springer-Verlag.

[Boo83] Abraham Bookstein. Information retrieval: A sequential learning process. *Journal of the American Society for Information Science*, 34(4):331–342, September 1983.

[Boo85] Abraham Bookstein. Implications of Boolean structure for probabilistic retrieval. In *Proceedings of the Eigth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 11–17. ACM Press, June 1985.

[BR84] J. D. Bovey and S. E. Robertson. An algorithm for weighted searching on a Boolean system. *Information Technology,* 3(2):84–87, April 1984.

[CL68] C.K. Chow and C.N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory,* IT-14(3):462–467, May 1968.

[Coo95] William S. Cooper. Some inconsistencies and misidentified modeling assumptions in probabilistic information retrieval. *ACM Transactions on Information Systems,* 13(1):100–111, January 1995.

[Cro86] W. Bruce Croft. Boolean queries and term dependencies in probabilistic retrieval models. *Journal of the American Society for Information Science*, 37(2):71–77, March 1986.

[CY82] D. Chow and Clement Yu. On the construction of feedback queries. *Journal of the Association for Computing Machinery*, 29:127–151, January 1982.

[EC92] Daniel M. Everett and Steven C. Carter. Topology of document retrieval systems. *Journal of the American Society for Information Science,* 43(10):658–673, 1992.

[Eva94] Ross Evans. Beyond Boolean: Relevance ranking, natural language and the new search paradigm. In *Proceedings of the Fifteenth National Online Meeting,* pages 121–128, Medford, NJ, 1994. Learned Information.

[FW90] Edward A. Fox and Sheila G. Winett. Using vector and extended Boolean matching in an expert system for selecting foster homes. *Journal of the American Society for Information Science,* 41(1):10–26, 1990.

[Gup87] Padmini Das Gupta. Boolean interpretation of conjunctions for document retrieval. *Journal of the American Society for Information Science*, 38(4):245–254, July 1987.

[Haa95] Stephanie W. Haas. Domain terminology patterns in different disciplines: Evidence from abstracts. In *Proceedings of the Fourth Annual Symposium on Document Analysis and Information Retrieval,* pages 137–146, Las Vegas, NV, April 1995.

[KK84] William Kneale and Martha Kneale. *The Development of Logic.* Clarendon Press, Oxford, 1984.

[LB88] Robert M. Losee and Abraham Bookstein. Integrating Boolean queries in conjunctive normal form with probabilistic retrieval models. *Information Processing and Management*, 24(3):315–321, 1988.

[LBY86] Robert M. Losee, Abraham Bookstein, and Clement T. Yu. Probabilistic models for document retrieval: A comparison of performance on experimental and synthetic databases. In *ACM Annual Conference on Research and Development in Information Retrieval*, pages 258–264, 1986.

[Lee94] Joon Ho Lee. Properties of extended Boolean models in information retrieval. In *ACM Annual Conference on Research and Development in Information Retrieval*, pages 182–190, New York, 1994. ACM Press.

[Lee95] Joon Ho Lee. Combining multiple evidence from different properties of weighting schemes. In *ACM Annual Conference on Research and Development in Information Retrieval*, pages 180–188, New York, 1995. ACM Press.

[LH95] Robert M. Losee and Stephanie W. Haas. Sublanguage terms: Dictionaries, usage, and automatic classification. *Journal of the American Society for Information Science*, 46(7):519–529, 1995.

[Los88] Robert M. Losee. Parameter estimation for probabilistic document retrieval models. *Journal of the American Society for Information Science*, 39(1):8–16, January 1988.

[Los91] Robert M. Losee. An analytic measure predicting information retrieval system performance. *Information Processing and Management*, 27(1):1–13, 1991.

[Los94a] Robert M. Losee. Term dependence: Truncating the Bahadur Lazarsfeld expansion. *Information Processing and Management*, 30(2):293–303, 1994.

[Los94b] Robert M. Losee. Upper bounds for retrieval performance and their use measuring performance and generating optimal Boolean queries: Can it get any better than this? *Information Processing and Management*, 30(2):193–203, 1994.

[Los95a] Robert M. Losee. Determining information retrieval performance without experimentation. *Information Processing and Management*, 31(4):555–572, 1995.

[Los95b] Robert M. Losee. The development and migration of concepts from donor to borrower disciplines: Sublanguage term use in hard & soft sciences. In Michael E. D. Koenig and Abraham Bookstein, editors, *Proceedings of the Fifth International Conference on Scientometrics and Informetrics*, pages 265–274, June 1995.

[Los96] Robert M. Losee. Evaluating retrieval performance given database and query characteristics: Analytic determination of performance surfaces. *Journal of the American Society for Information Science*, 47(1):95–105, 1996.

[LY82] K. Lam and C. T. Yu. A clustered search algorithm incorporating arbitrary term dependencies. *ACM Transactions on Database Systems*, 7:500–508, 1982.

[MC81] Karen Markey and Pauline Atherton Cochrane. *ONTAP: An Online Traininga nd Practice Manual for ERIC Database Searchers*. ERIC Clearinghouse on Information Resources, Syracuse U., NY, 2nd edition, October 1981.

[Moo93] Sung Been Moon. *Enhancing Retrieval Performance of Full-Text Retrieval Systems Using Relevance Feedback*. PhD thesis, U. of North Carolina, Chapel Hill, NC, 1993.

[Nie89] Jianyun Nie. An information retrieval model based on modal logic. *Information Processing and Management*, 25(5):477–491, 1989.

[Por80] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.

[Rad79] Tadeusz Radecki. Fuzzy set theoretical approach to document retrieval. *Information Processing and Management*, 15:247–259, 1979.

[Rad82] Tadeusz Radecki. A probabilistic approach to information retrieval in systems with Boolean search request formulations. *Journal of the American Society for Information Science*, 33(6):365–370, November 1982.

[Rob77] Stephen E. Robertson. The probability ranking principle in IR. *Journal of Documentation*, 33(4):294–304, 1977.

[RSJ76] Stephen E. Robertson and Karen Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information Science*, 27:129–146, 1976.

[RT90] S. E. Robertson and C. L. Thompson. Weighted searching: the CIRT experiment. In *Informatics 10: Prospects for Intelligent Retrieval*, pages 153–166. ASLIB, London, 1990.

[RTMB86] S. E. Robertson, C. L. Thompson, M. J. Macaskill, and J. D. Bovey. Weighting, ranking and relevance feedback in a front-end system. *Journal of Information Science*, 12(1/2):71–75, 1986.

[Sag81] Naomi Sager. Information structures in texts of a sublangauge. In *Proceedings of the 44th ASIS Annual Meeting*, pages 199–201, White Plains, NY, 1981. Knowledge Industry Publications.

[Sal84] Gerard Salton. The use of extended Boolean logic in information retrieval. Technical Report TR 84–588, Cornell University, Computer Science Dept., Ithaca, N.Y., January 1984.

[Sme84] A. F. Smeaton. Relevance feedback and a fuzzy set of search terms in an information retrieval system. *Information Technology: Research and Development*, 3(1):15–23, January 1984.

[Spi95] Amanda Spink. Term relevance feedback and mediated database searching: Implications for information retrieval practice and systems design. *Information Processing and Management*, 31(2):161–171, 1995.

[Tib92] Helen R. Tibbo. Abstracting across the disciplines: A content analysis of abstracts from the natural sciences, the social sciences, and the humanities with implications for standardization and online information retrieval. *Library and Information Science Research*, 14:31–56, 1992.

[Tur94] Howard Turtle. Natural language vs. Boolean query evaluation: A comparison of retrieval performance. In *ACM Annual Conference on Research and Development in Information Retrieval*, pages 212–220, New York, 1994. ACM Press.

[VR77] C.J. Van Rijsbergen. A theoretical basis for use of co-occurrence data in information retrieval. *Journal of Documentation*, 33(2):106–119, June 1977.

[YBLS83] Clement T. Yu, Chris Buckley, K. Lam, and Gerard Salton. A generalized term dependence model in information retrieval. *Information Technology: Research and Development*, 2(4):129–154, 1983.