Percent Perfect Performance (PPP)

Information Processing & Management, 43 (4), 2007, 1020-1029

Robert M. Losee*
CB#3360
University of North Carolina
Chapel Hill, NC 27599-3360

email: *losee at unc period edu*

March 25, 2007

Abstract

An information retrieval performance measure that is interpreted as the percent of perfect performance (PPP) can be used to study the effects of the inclusion of specific document features or feature classes or techniques in an information retrieval system. Using this, one can measure the relative quality of a new ranking algorithm, the result of incorporating specific types of metadata or folksonomies from natural language, or determine what happens when one makes modifications to terms, such as stemming or adding part-of-speech tags. For example, knowledge that removing stopwords in a specific system improves the performance 5% of the way from the level of random performance to the best possible result is relatively easy to interpret and to use in decision making; using this percent based measure also allows us to simply compute and interpret that there remains 95% of the possible performance to be obtained using other methods. The PPP measure as used here is based on the Average Search Length, a measure of the ordering quality of a set of data, and may be used when evaluating all the documents or just the first N documents in an ordered list of documents. Because the ASL may be computed empirically or may be estimated analytically, the PPP measure may also be computed empirically or performance may be estimated analytically. Different levels of upper bound performance are discussed.

1 Introduction

Many classic studies of information retrieval examined the contributions made to retrieval performance through use of various kinds of document features

*The author wishes to thank Lewis Church, Lee Roush, and two anonymous referees for comments on an earlier version of this article.

and system characteristics (Cleverdon, 1967; Salton & Lesk, 1968). This form of study of the inclusion of different types of system characteristics, such as controlled vs. uncontrolled vocabularies, the relative utility of including terms of class *nouns* vs. *verbs*, stemmed vs. unstemmed terms, and so forth, continues to the present. The measure we propose here captures the percent of the upper bounds performance provided by using a particular retrieval system options or characteristics, and is referred to here as the *Percent of Perfect Performance (PPP)*. The PPP measure is inspired by the statistical R^2 value, which measures the extent to which information about one variable predicts the variation in another variable and is scaled from zero to one hundred percent. Similarly, the PPP measure computes the percent of the optimal ranking performance that is provided by the system being studied, providing information about whether the performance is nearly optimal, or perhaps it is only slightly above random. If a system using a certain method has a PPP value of 10%, it achieves 10% of the possible performance, which also implies that there remains 90% of the possible performance to be obtained using other methods. The PPP method may be computed retrospectively based upon existing documents and relevance judgments, or it may be predicted analytically, based upon parameter values.

The performance of ordering systems that are used to retrieve documents can be measured using a number of popular measures, including precision and recall (Salton & McGill, 1983), expected search length (Cooper, 1968), and the closely related E and F measures (Swets, 1969; Van Rijsbergen, 1974; Shaw, 1986). Many of these measures can be shown to have strong relationships (Demartini & Mizzaro, 2006; Egghe, 2004; Losee, 2000). More recently, measures have been explicitly designed to work without full knowledge of relevance for all documents (Buckley & Voorhees, 2004).

Using the Average Search Length (ASL) measure of retrieval system performance, the average position of relevant documents in the ordered list of documents, has some benefits over the other measures in that it can be analytically predicted in some cases or measured empirically in all cases, it explicitly addresses tied document weights, and its interpretation is simple (Losee, 2000, 2006). The ASL serves as the basis for the PPP performance measure developed below. Positions in the ordered list of documents are numbered so that position 1 is the location of the first document in the list and position N is the location of the last document in the list. As the average position of relevant documents in the ordered list of documents, a low value, approaching 1, represents the average position of relevant documents being near the front of the ordered list of documents, while a high value, approaching N, the number of documents in the ordered list, represents the average position of relevant documents being near the end of the ordered list. Given the ordered list of documents, $r, n, r, n, n,$ with r here representing a relevant document and n representing a non-relevant document, and the documents are strongly ordered from left to right, then the two relevant documents, at positions 1 and

3, would produce an ASL of $(1 + 3)/2 = 2$, with position 2 being determined to be the average position of relevant documents When weak ordering occurs and several successive documents have the same document weight, the position used in computing the ASL for each of the documents with equal weights is the average position for those documents with that given weight. As the expected position of a relevant document, this can be computed from data, as in the example above, or the ASL can be predicted analytically.

The Average Search Length can be applied to an entire dataset or to a portion of it. While the empirical results below examine the retrieval performance of the entire ordered list of documents, the ASL may also be computed for a portion of the ordered list, usually starting at the beginning of the ordered set and moving in a fixed distance. For example, one may compute the ASL for only the first ten document or the first one hundred documents. This provides an effective measure of performance when high precision searches are being studied or when relevance is available for only an initial set of the retrieved documents.

The ASL may be normalized different ways to produce a Normalized Average Search Length ($NASL$), depending on the desired characteristics of the NASL (Losee, 2006). The measure $NASL$ is the percent of all documents that are ranked ahead of the average position of the relevant documents and thus the probability that a randomly selected document will be ranked ahead of the average position of relevant documents. Just as their are numerous methods for estimating a probability or probability distribution, such as Bayesian methods, maximum likelihood, or method of moments, the $NASL$ may similarly be estimated using several different approaches, all of which are reasonable. The ASL can be normalized by dividing by N for N values in the hundreds or greater, yielding the $NASL$. Here N is the number of documents being studied, which might be all documents or just the first ten or one hundred documents. Note that dividing ASL by N yields an $NASL$ that can have the value 1 but never reaches down to 0.

The $NASL$ can be computed empirically from the ASL by scaling the ASL, whose range of values for N documents is 1 to N down to the range of 0 to 1 by computing $NASL$ as $(ASL - 1)/(N - 1)$. While this formulation produces an $NASL$ that ranges from 0 to 1, we may choose to have an $NASL$ that has a slightly different range. If $N = 3$ and we have 3 different ASL values, 1, 2, and 3, the lowest might be placed in the middle of the bottom $1/3$, that is, at $1/6$, while the second value would be in the middle of the middle third (e.g., at point $3/6$), and the highest value in the middle of the top third (e.g., at point $5/6$, halfway between $4/6$ and $6/6$). We can achieve this with

$$NASL = (ASL - 1/2)/N. \tag{1}$$

Thus, for 3 ASL positions of 1, 2, and 3, the three $NASL$ values will be $1/6$, $3/6$, and $5/6$. Equation 1 is used in the experimental results below.

One can compute a value related to $NASL$, $\mathcal{W}$, as the percent of documents in the first half of the ranked list that are ahead of the average position of relevant documents (with the average computed from the entire list) (Losee, 2006). This may be described as the probability that a document from the top half of the ranked list of documents is ranked ahead of the expected position of a relevant document. If we assume that the $NASL$ is one half or less, that is, the performance is equal to or better than random, then we may compute $\mathcal{W} = 2 \times NASL$.

Documents from the first half of the list are used instead of the entire list because it is desirable that the measure focus on the positive aspects of retrieval, occurring when the average position of a relevant document is located in the first half of the ordered list of documents. Using this positive section of the ordered list of documents is most easily enabled by assuming that the ranking method is better than random, and then multiplying the probability that a document is ahead of the expected position of relevant documents by 2. The worst case value for $\mathcal{W}$ is then 1, which occurs when random ordering occurs, and the best case value is then $\mathcal{W} = 0$, when the expected position of relevant documents is at the front of the ordered list of documents. Achieving this range of 0 to 1 for $\mathcal{W}$ is the primary motivation for this computation of $\mathcal{W}$.

The development of PPP was originally based on the outgrowth of a quantitative measure (and resulting qualitative analysis) of how a single feature would contribute toward improving retrieval performance. The *Relative Feature Utility* (RFU) is computed from the number of features (e.g., terms) or feature sets of one type whose use produces equivalent performance to using a single feature or feature set of another type (Losee, 2006). A *feature* here is a characteristic of a document that the system developer decides to incorporate into ranking or ordering algorithms. A document might contain the term *the* or it might be be absent; this is a binary feature that is present or absent. A system may be implemented so as to include a feature or characteristic when the system developer or manager decides that such a feature should be included when calculating query and document similarity; many system managers in English speaking countries are likely to decide that a word with little meaning such as *the* might be excluded from the set of features to be used in document ranking calculations by the retrieval system. To simplify discussion here, we assume that all features are binary and that relevance is binary, although non-binary feature frequencies and continuous relevance are easily incorporated into calculating $NASL$ and thus $\mathcal{W}$ (Losee, 1998, 2006). Given two binary features, i and j, where i might be used to represent the presence or absence of the term *information* and j might be used to represent the presence or absence of the term *document*, $\mathcal{W}_i$ and $\mathcal{W}_j$ are the performance probabilities associated with features i and j. One may compute the number of systems with $\mathcal{W}_j$ performance probabilities that produce the same numeric value as a system with performance probability $\mathcal{W}_i$ by solving $\mathcal{W}_i = \mathcal{W}_j^{\mathcal{M}}$ for $\mathcal{M}$, which gives us Equa-

tion 2 below. We assume that all the $\mathcal{W}_i$ values are independent and identically distributed, as are the $\mathcal{W}_j$ values in the analysis below. Denoted as $\mathcal{M}$, the relative feature utility of system type i compared to system type j is

$$\mathcal{M} = \frac{\log(\mathcal{W}_i)}{\log(\mathcal{W}_j)} \tag{2}$$

and indicates that there are $\mathcal{M}$ occurrences of statistically independent type j features that together give us the performance associated with using a single type i.

As an example, consider the number n of coin tosses, each toss with probability of $1/2$, that result in the same probability as achieving a specific roll on a sixteen sided die. We could express this as algebraically solving the equation $(1/2)^n = 1/16$ for n. In this case $n = 4$. Those unfamiliar with logarithms might note that $\log_2 8$ asks the question, "2 to what power is 8" and the answer is clearly 3, since $2^3 = 8$. Given a problem such as $16 = 4^{\mathcal{M}}$, we find that, using logarithms computed to (arbitrarily chosen) base 2, we may solve this as $\mathcal{M} = \log_2 16 / \log_2 4$. Because 2 to the fourth power is 16, and 2 to the second power is 4, this becomes $\mathcal{M} = 4/2 = 2$, which is consistent with our knowledge that, going back to the earlier problem, $16 = 4^{\mathcal{M}}$, and $16 = 4^2$.

In many decision making situations, it is desirable to be able to compare the relative utility of various options. For example, a single noun might be expected to have the same ordering capability as 1.5 or 2 adjectives (Losee, 2006). $\mathcal{M}$ is easily interpreted and can be used in other computations, such as our measure of the percent of possible performance that is provided by using a particular option set. Consider a situation where $NASL_i = 0.43$ and $NASL_j = 0.48$. The value $\mathcal{M}$ is then computed as $\mathcal{M} = \log(2 \times 0.43) / \log(2 \times 0.48) = 3.69$. This implies that retrieval option i will result in 3.69 times the performance that will be found with retrieval option j, or that it will take 3.69 features of type j to perform as well as a single feature of type i.

2 Percent Perfect Performance $\mathcal{P}$

The performance of a retrieval system may be compared to the level of random performance and of upper bounds performance by first using the ASL measure to measure retrieval performance, whether of an entire system or when studying specific features, and then performing further calculations to produce the performance value. When comparing the relative feature utility performance achieved using document ordering system x with the relative feature utility performance obtained at the upper bounds, we can compute the Percent Perfect Performance (PPP). Such a value can provide a measure of the relative percent of achievable performance above the random performance provided by a situation. For example, we might say that using only nouns or using folksonomies provides performance that is 10% of the way from random per-

formance to the upper bounds, leaving another 90% to be achieved using additional methods.

Using the RFU and Equation 2 above, one may compute the improvement of a system i over a baseline system c and the improvement of the upper bounds system u over the same baseline system. The percent performance $\mathcal{P}_i^u$ of system i in the context of the upper bounds system u may be computed by dividing the appropriate $\mathcal{M}$ values, the number of c values equivalent to system i, normalized by dividing by the number of c values in the upper bounds system, based on an arbitrary constant base system c, as

$$\mathcal{P}_i^u = \frac{\log \mathcal{W}_i / \log \mathcal{W}_c}{\log \mathcal{W}_u / \log \mathcal{W}_c} = \frac{\log \mathcal{W}_i}{\log \mathcal{W}_u}. \tag{3}$$

The feature or system c may be interpreted as a baseline performance level, with the numerator (or denominator, respectively) in Equation 3 showing the number of baseline systems that are performatively equivalent (have the same numeric performance value) to system i (or the upper bounds, respectively). When the number of baseline systems performatively equivalent to system i is divided by the number possible (the upper bounds), the percent of the upper bounds performance provided by system i's performance is produced. The $\mathcal{P}$ values may be multiplied by 100 to provide the percent of perfect performance (PPP) provided by a system compared to the upper bounds.

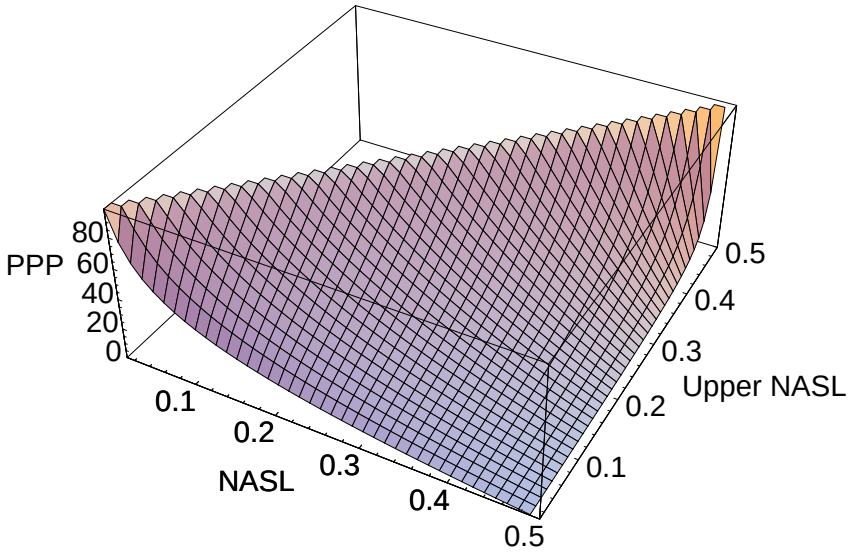
It was noted above that when the $\mathcal{M}$ values are computed from the random level of performance using Equation 3, $NASL = .5$, then $\mathcal{M}$ is infinite as $(\log x)/(\log 1) = \infty$ because we are dividing by $\log 1 = 0$. However, using the ratio in Equation 3, the relative merit of the different $\mathcal{M}$ values may be computed so that the base points, whether they are $NASL = 0.5$ or another value $NASL = c$ which might be very close to random (e.g., $NASL = 0.4999$), will cancel out so that the portion of the $\mathcal{M}$ value for the upper-bounds due to x may be computed. Thus, the performance using a ranking algorithm and features denoted as x given upper bounds performance u, is suggested by the right side of Equation 3 as

$$\mathcal{P}_x^u = \frac{\mathcal{M}_x}{\mathcal{M}_u} = \frac{\log(2 \times NASL_x)}{\log(2 \times NASL_u)}.$$

Note that if the upper bound is arbitrarily set to $NASL$ for system x as .1, for example, then the best level of performance $x = .1$ produces $\mathcal{P}_{.1}^{.1} = \log(2x)/\log(2u) = 1$ or 100%. Similarly, for $x = .5$ and the upper bounds remains at .1, $\mathcal{P}_{.5}^{.1} = \log(2 \times 0.5)/\log(2 \times 0.1) = 0$ or 0% and for $x = .3$, $\mathcal{P}_{.3}^{.1} = \log(2 \times 0.3)/\log(2 \times 0.1) = 0.318$. The latter result may be interpreted as implying that performance at level x is almost 32% of the possible level of performance obtained with a perfect ordering procedure.

While one may compute $\mathcal{P}$ using the feature model proposed above, it may also be computed more directly from the ASL values, with a focus being more

Figure 1: Percent Perfect Performance (PPP) for a range of $NASL$ and upper bound $NASL$ values.



on the performance measure (e.g., ASL) rather than the document model (e.g., features). This will be examined more below.

Figure 1 shows the performance that is obtained over a range of $NASL$ values and a range of upper bounds $NASL$ values. This non-linear relationship shows that a high performance value is obtained when the $NASL$ being studied approaches the upper bounds.

3 Ordering Performance Measures

A range of performance measures have been developed to evaluate the ordering of documents, given a range of different considerations (Harter & Hert, 1997; Demartini & Mizzaro, 2006). Most traditional ordering performance measures assume the existence of a relevance judgment for each document. The most popular performance measures are probably precision, the percent of documents retrieved that are relevant, and recall, the percent of relevant documents in the database that have been retrieved. Combinations of precision and recall may be used as single number measures of performance at various points in the search process. The F measure is related to the harmonic mean of precision and recall and may be studied at a specific point in the document

retrieval process.

Some other single number measures of ordering performance address the length of time or the number of documents that are examined when moving to a specific place in the ordered set of documents. The Expected Search Length (ESL) measures the number of non-relevant documents occurring before a specific point in the ordered list of documents (Cooper, 1968), while the Average Search Length (ASL) measures the number of documents encountered as one moves to the average position of relevant documents.

As search engines have become popular and the size of retrieval databases grows from thousands to millions and to billions (thousands of millions), an increasing number of the searches conducted are high-precision searches where a few useful documents are desired. Most of the measures discussed above require that relevance judgments be available for all relevant documents, a practical difficulty when there is a very large number of documents. A popular measure, the Mean Average Precision (MAP), measures the precision after each relevant document is retrieved and then averages these precision values (Buckley & Voorhees, 2004; Jarvelin & Kekalainen, 2002; Sanderson & Zobel, 2005; Voorhees, 2001). MAP requires relevance judgments for only those documents up to the point at which the MAP is computed, often the documents that are retrieved in a high precision search, or up to a specific point in a search. It becomes more and more difficult to locate and retrieve relevant documents as searches progress, so precision is highest at first and decreases as a search progresses. The more the relevant documents are located near the beginning of the ordered list of documents, the higher will be the MAP. Given different levels of relevance beyond the simple binary relevant vs. non-relevant values, measures expanding beyond the MAP approach may be used, such as the discounted cumulative gain, which considers more than two relevance levels and discounts the value of a document the further the document is from the beginning of the list (Jarvelin & Kekalainen, 2002; Voorhees, 2001).

Because ASL and $NASL$ may be computed from an initial set of retrieved documents, or from all the documents, one may compute the $\mathcal{P}$ measure from either all the documents or from the set of documents of a certain size at the beginning of the ordered list. In this way, the $\mathcal{P}$ measure can be used to study high precision searches, as do the MAP and cumulative gain measures.

The PPP ($\mathcal{P}$) performance measure that was developed in the previous section acts as a single number measure of ordering performance. While developed here to measure the percent performance toward optimal performance, with the ASL as the basic performance measure, this same technique is also applicable to other performance measures that are probabilities or average probabilities.

4 Normalized Measures vs. $\mathcal{P}$

Measures are often normalized to place them in a range of 0 to 1, 0 to 100, or some similar range that allows for simple interpretations and comparison of different values. Normalizing a measure frequently occurs by taking the value v and the range of the possible values, from the lowest value v_ℓ to the highest value v_h, and then dividing the degree to which the value being examined exceeds the lowest value; this is normalized by (divided by) the range of possible values:

$$Normalized(v) = \frac{v - v_\ell}{v_h - v_\ell}.$$

The normalized v value is thus in the range of 0 to 1, with $Normalized(v_\ell) = 0$ and $Normalized(v_h) = 1$.

Normalized measures of performance are particularly useful when studying performance with different length measures. The author has found a normalized version of the ASL measure useful for comparing different performance values, as well as for studying the normalized upper bounds performance. Cooper has advocated the use of a normalized form of the expected search length and this may be the most popular normalized search length measure (Cooper, 1968).

The $\mathcal{P}$ measure functions as a percent of optimal performance and is developed explicitly to be a linear percent, normalized so as to range from 0% to 100%, which is the same as a probability of 0 to 1. Returning to our earlier explanation of the basis of the Relative Feature Utility, one can compute the number of occurrences, $\mathcal{M}$, of system type j events with performance probability p_j that is equivalent to the probability of a single system event of type i with performance probability p_i is computed by first noting that $p_i = p_j^{\mathcal{M}}$ (or, similarly, $p_i^{1/\mathcal{M}} = p_j$). Solving algebraically for $\mathcal{M}$ produces an equation like Equation 2.

One can similarly solve for the $\mathcal{M}$ associated with the upper bounds of performance. Somewhat differently than with the RFU, the $\mathcal{M}$ used in our $\mathcal{P}$ measure is the fraction of an occurrence of a type j event, instead of the number of occurrences as in the RFU. Thus, the PPP measure $\mathcal{P}$ is computed as the percent of the occurrence of the upper bounds performance that produces the performance associated with a single event of type i, the performance being studied. Thus, $\mathcal{M}$ will be a fraction value representing the fraction of the upper bounds. Continuing with the notation above, and denoting the upper bound performance as type u with probability p_u, we solve for $\mathcal{M}$ in $p_i = p_u^{\mathcal{M}}$, yielding Equation 3 above. Here the probabilities p represent $\mathcal{W}$ values, the probability that a document in the top half of the documents is ahead of the average position of a relevant document. Note that we could not solve for $\mathcal{M}$ if the $\mathcal{W}$ values were not probabilities.

5 Upper Bounds

There are several different upper bounds, or levels of maximum performance, that may be used in computing $\mathcal{P}$ values. Starting at the highest possible upper bounds, an oracle might be able to look at media and determine whether they are relevant or not relevant to a user's query, regardless of the terms that might be in the query and regardless of the degree to which the query and the documents' terms match. For example, two documents might have the same feature profile visible to the searcher, while one is considered *relevant* by a user and the other labeled *non-relevant*. Ranking by this omniscient level of upper bounds would place the relevant document before the non-relevant document. We denote the performance given options x and given this type of upper bounds as $\mathcal{P}_x^\infty$.

The best possible upper bounds would occur when the R relevant documents in the list of documents being studied occur at the beginning of the list of documents before any non-relevant documents occur. The $NASL$ value of this may be computed by noting that the expected position of a relevant document is at $R/2 + 1/2$, which is the best case ASL. Using this ASL, we may compute the upper bounds $NASL$ ($NASL_\infty$) using Equation 1:

$$NASL_\infty = \frac{(R/2 + 1/2) - 1/2}{N} = \frac{R/2}{N}. \tag{4}$$

Those without access to omniscience might have a lower level of upper bounds in which the relevance of a document is fully determined by the features present or absent in the documents and the feature space that is used and thus knowing the document's features, such as whether it has particular terms, along with the full set of possible features, allows one to definitively determine the document's relevance. Upper bounds performance at level x for this level of upper bounds is denoted as $\mathcal{P}_x^{AllTerms}$ This upper bounds performance is expected to be worse than or equal to the upper bounds performance provided by using the oracle described above, and thus $\mathcal{P}_x^{AllTerms} \geq \mathcal{P}_x^\infty$.

Given a large number of terms or features being used in the ordering, the best possible ordering will be very similar, if not identical to, the best possible ordering described above, $\mathcal{P}_x^\infty$. If we consider b binary features, there are about 2^b possible sets of document characteristics, and given even a small vocabulary of, for example, 50 terms, one would find so many sets of characteristics that, if the terms were approximately evenly distributed, the $NASL_\infty$ would usually be very closely approximated by $NASL_{AllTerms}$, and thus $\mathcal{P}_x^{AllTerms} \approx \mathcal{P}_x^\infty$.

Documents may be ordered based solely on the features present in the query, with the document weights being then determined by the presence or absence of the query terms in the documents. Empirical performance at level x in this case is denoted as $\mathcal{P}_x^{Query}$. In the case where there are more terms in the feature space than there are in the query, then the upper bounds performance

using only query terms is worse than the level of performance obtained when using all terms in the features space. In the special case where the query has all the features in the feature space, then the performance $\mathcal{P}_x^{AllTerms} = \mathcal{P}_x^{Query}$.

In many cases, we can determine what percent of the contribution to performance is provided by incorporating an intermediate system feature i when considering the performance of final system feature f. Beginning with direct performance $\mathcal{P}_f^\infty$ and the performance $\mathcal{P}_{f,i}^\infty$, the performance improvement due to adding feature or system component i as part of computing performance at level f is computed as $\Delta_{\mathcal{P}_{i|f}^\infty} = \mathcal{P}_{f,i}^\infty - \mathcal{P}_f^\infty$. For example, if the performance with feature f by itself is 12% of the way to perfection and it improves to 15% when using feature i, we can conclude that feature i improved performance by 3%.

6 How to Apply PPP—An Example

As an example of how to apply PPP, consider an ordered list of documents with r or n, denoting relevance or non-relevance, respectively, as well as the profile of features that are suggested for use based on the query, with the features shown in binary as a subscript:

$$r_{10}, n_{10}, r_{10}, n_{11}, n_{00}, r_{01}, n_{01}.$$

When computing Average Search Length, documents with equal profiles are treated as they are all located at the center position of the equally profiled documents. The ASL for this ordered set of documents is thus 2 relevant documents at position 2 (the center of the first 3 documents), and a relevant document at position 6.5, the middle of positions 6 and 7 for r_{01} and n_{01}. The ASL is thus positions $(2 + 2 + 6.5)/3 = 10.5/3 = 3.5$ From this, we may compute the $NASL$ using Equation 1 as $NASL = (3.5 - .5)/7 = 3/7 = 0.43$.

To compute the upper bounds of performance, we will reorder documents so they are in weakly decreasing order by the average precision of the documents with a given profile. Thus, the upper bounds ordering becomes

$$r_{10}, n_{10}, r_{10}, r_{01}, n_{01}, n_{11}, n_{00}.$$

The ASL for this upper bounds ordering is computed from 2 documents at position 2 and 1 document at position 4.5, thus $ASL = (2+2+4.5)/3 = 8.5/3 = 2.833$. The $NASL$ is computed as $NASL = (2.833 - .5)/7 = 2.33/7 = .33$. Note that we could compute the $NASL_\infty$ as $(R/2)/N$ or $(3/2)/7 = 3/14 = .21$.

We may compute the Percent Perfect Performance by using Equation 3. We find that $\mathcal{P} = \log(2 \times .43)/\log(2 \times .33) = \log .86/\log .66 = .36$ We can thus state that our original ordering method and these documents achieved 36% of the performance that is possible, leaving another 64% to be obtained, with performance measured by ASL.

The utility of such a number is more obvious when we have two different

ranking procedures or retrieval options. Given the data from earlier in this section, if assigning part-of-speech tags to the document profiles allowed us to separate the two relevant documents with profile 10 from the non-relevant document with profile 10, the ranking technique is likely to produce better rankings. Note that if we consider part-of-speech tags as an option for our system, the upper bounds must be computed so as to be consistent with this improved performance. Using part-of-speech tags would improve some of the upper bound $NASL$ measure of performance.

The following section shows the comparison of different retrieval options for some standard test databases and several well understood techniques, such as part-of-speech tagging and term stemming.

7 Empirical Results

Several tests were conducted that provide results measured using the PPP $\mathcal{P}$ measure. Rankings were consistent with the CLMF (Coordination Level Matching – Frequency) weight, a modification of Coordination Level Matching in which the number of terms in the document that are also in the query are counted (Losee, 2006). This is like the TF IDF (Term Frequency times Inverse Document Frequency) weighting except that all terms are given the same term weight. One should note that unlike IDF (Inverse Document Frequency) weighting which would give common words such as stopwords (common, non-subject bearing terms) a very low weight, CLMF gives stopwords an equal weight as other terms and thus ridding documents of stopwords becomes more effective in our results than would be found with TF IDF weighting.

The CLMF provides a simple, easy to understand weight (simpler than TF IDF). Our concern in this (and most other studies) is simplicity and clarity, rather than achieving the best results possible. Increasing the reader's understanding of what occurs and the nature of the relationships between system variables is the goal of many scientists; clearly, achieving the best results possible is also a valid goal for researchers.

Two standard databases are used in producing these measurements on the Nyltiac (http://Nyltiac.com) retrieval system. The first 50 queries of the CF database, composed of documents with the subject heading Cystic Fibrosis (CF) in the National Library of Medicine database, are used (Wood, Wood, & Shaw, 1989; Moon, 1993) and the parts-of-speech (POS) tags are supplied by the Brill tagger (Brill, 1994) for the CF POS Tagged database. The MED1033 database, composed of 1033 documents extracted from the National Library of Medicine's database, was used for other analyses. Because of the design of this database, it is easier for systems using MED1033 to retrieve documents labeled as relevant (Kwok, 1990; Shaw, Burgin, & Howell, 1997). The MED1033 Tagged database is part-of-speech tagged as with the CF POS Tagged database. The stopword list used here contains 425 stopwords, and removing stopwords may

have a larger impact on performance in this study than when using smaller lists of stopwords containing only a few dozen terms.

Table 1: Measures of retrieval performance (PPP) showing the percent of the upper bounds performance level that is obtained with the indicated processing options.

Description	*CF*		*CF POS Tagged*		*MED1033*		*MED1033 Tagged*	
	$NASL$	$\mathcal{P}^{\infty}$	$NASL$	$\mathcal{P}^{\infty}$	$NASL$	$\mathcal{P}^{\infty}$	$NASL$	$\mathcal{P}^{\infty}$
Full Query	0.4192	5.67%	0.4182	5.31%	0.3751	8.70%	0.3722	8.98%
Case Sensitivity Removed	0.418	5.51%	0.4215	5.15%	0.3751	8.70%	0.3722	8.98%
Case & Stems Removed	0.4046	6.13%	0.4216	5.16%	0.3612	9.70%	0.3717	9.02%
Case & Stopwords Removed	0.2986	14.21%	0.3151	12.91%	0.1261	43.16%	0.1332	42.02%
Case, Stopwords, & Stems Removed	0.2851	15.31%	0.3146	13.18%	0.1223	42.96%	0.1312	42.25%
Above & POS tags	0.3146	13.18%	0.3146	13.18%	0.1312	42.25%	0.1312	42.25%
Above & Only Nouns & Adj	0.3132	13.33%	0.3132	13.33%	0.1293	42.79%	0.1293	42.79%
Upper Bounds	0.0042	100.00%	0.0042	100.00%	0.0112	100.00%	0.0112	100.00%

Results given in Table 1 show the upper bounds (at the bottom of the table) having $\mathcal{P}$ levels of 100%. With case ignored and stems and stopwords removed, the Percent of Perfect Performance was about 15% for the CF database and about 43% for the MED1033 database (both untagged). Using part-of-speech tags produces slightly lower results. This data suggests that the greatest contribution is provided by removing stopwords, while most of the other options provide little improvement. Knowing that removing the stopwords and case sensitivity produces about 15% of the possible performance for CF and about 43% of the possible performance for the MED1033 database gives us an idea as to whether we should include this kind of processing; a 15% improvement is probably enough to justify most system designers to incorporate an option. The 43% performance improvement leaves us 57% of performance improvement remaining to be addressed by other methods.

8 Discussion and Conclusions

Using the $\mathcal{P}$ measure of Percent of Perfect Performance (PPP), the percent of the possible (upper bounds) performance accounted for by using the current system and document features, we have been able to illustrate how one might measure the contribution to performance of several information retrieval options. Using such a measure allows us to understand the relative utility of different features in terms of percent improvement toward optimality, a value that most searchers may understand with little training, which is not the case for most other retrieval measures. By comparing the performance for two systems, with one being the upper bound, we are able to compute the percent of the upper bound performance of one system that is provided by the performance of a non-optimal system. We believe that this simplicity and comparison to upper bounds is advantageous to both researchers, searchers, and to those making decisions about the use of systems in organizational contexts.

Empirical results were provided showing the application of the PPP measure using traditional retrieval test databases. As information retrieval matures as a science and becomes more analytic, the ability to predict retrieval performance becomes increasingly important. As the ASL and $NASL$ values may be predicted analytically, one may predict PPP performance based on analytic considerations. For example, one might be interested in predicting what the performance curve looks like as the percent of relevant documents in a database increase, or as the difficulty in locating these documents increases; these performance results, presented as percents of upper bounds performance, may be produced with relatively little effort using graphic packages.

We should note that when comparing our results to the theoretical upper bounds, we do not wish to imply that all or even many humans can provide ranking at the $\mathcal{P}^{\infty} = 100\%$ level. Future work might examine the optimal

performance that is routinely achievable by human analysis and the sorting of documents, and what the parameters are for this level of performance. Information retrieval performance might then be determined based on what percent of the performance achievable by most humans (H), $\mathcal{P}_x^H$, may be provided by methods x. Information retrieval systems might have as their goal ordering documents at the level of humans, or scholars may wish to surpass human performance if it is significantly below $\mathcal{P}^\infty$.

References

Brill, E. (1994). Some advances in transformation-based part of speech tagging. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, pp. 722–727 Menlo Park, CA. AAAI Press.

Buckley, C., & Voorhees, E. M. (2004). Retrieval evaluation with incomplete information. In Sanderson, M., Jarvelin, K., Allan, J., & Bruza, P. (Eds.), *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, England*, pp. 25–32 New York. ACM Press.

Cleverdon, C. (1967). The Cranfield tests on index language devices. *Aslib Proceedings*, *30*, 172–181.

Cooper, W. S. (1968). Expected search length: A single measure of retrieval effectiveness based on weak ordering action of retrieval systems. *Journal of the American Society for Information Science*, *19*(1), 30–41.

Demartini, G., & Mizzaro, S. (2006). A classification of IR effectiveness metrics. In *Advances in Information Retrieval: Lecture Notes in Computer Science, 3936*, pp. 488–491. Springer-Verlag, Berlin.

Egghe, L. (2004). A universal method of information retrieval evaluation: The missing link M and the universal IR surface. *Information Processing and Management*, *40*, 21–30.

Harter, S. P., & Hert, C. A. (1997). Evaluation of information retrieval systems: Approaches, issues and methods. In Williams, M. (Ed.), *Annual Review of Information Science and Technology*, Vol. 32, pp. 1–94. American Society for Information Science, Washington, D.C.

Jarvelin, K., & Kekalainen, J. (2002). Cumulative gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, *20*(4), 422–446.

Kwok, K. L. (1990). Experiments with a component theory of probabilistic information retrieval based on single terms as document components. *ACM Transactions on Information Systems*, *8*(4), 363–386.

Losee, R. M. (1998). *Text Retrieval and Filtering: Analytic Models of Performance*. Kluwer, Boston.

Losee, R. M. (2000). When information retrieval measures agree about the relative quality of document rankings. *Journal of the American Society for Information Science*, *51*(9), 834–840.

Losee, R. M. (2006). Is 1 noun worth 2 adjectives? Measuring the relative feature utility. *Information Processing and Management*, *42*(5), 1248–1259.

Moon, S. B. (1993). *Enhancing Retrieval Performance of Full-Text Retrieval Systems Using Relevance Feedback*. Ph.D. thesis, U. of North Carolina, Chapel Hill, NC.

Salton, G., & Lesk, M. (1968). Computer evaluation of indexing and text processing. *Journal of the ACM*, *15*(1), 8–36.

Salton, G., & McGill, M. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill, New York.

Sanderson, M., & Zobel, J. (2005). Information retrieval system evaluation: Effort, sensitivity, and reliability. In Marchionini, G., Moffat, A., Tait, J., Baeza-Yates, R., & Ziviani, N. (Eds.), *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil*, pp. 162–169 New York. ACM Press.

Shaw, Jr., W. M. (1986). On the foundation of evaluation. *Journal of the American Society for Information Science*, *37*(5), 346–348.

Shaw, Jr., W. M., Burgin, R., & Howell, P. (1997). Performance standards and evaluations in IR test collections: Cluster based retrieval models. *Information Processing and Management*, *33*(1), 1–14.

Swets, J. A. (1969). Effectiveness of information retrieval methods. *American Documentation*, *20*(1), 72–89.

Van Rijsbergen, C. (1974). Foundation of evaluation. *Journal of Documentation*, *30*(4), 365–373.

Voorhees, E. M. (2001). Evaluation by highly relevant documents. In Kraft, D. H., Croft, W. B., Harper, D. J., & Zobel, J. (Eds.), *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, Louisiana*, pp. 74–82 New York. ACM Press.

Wood, J. B., Wood, R. E., & Shaw, W. M. (1989). The cystic fibrosis database. Tech. rep. 8902, University of North Carolina, School of Information and Library Science, Chapel Hill, N.C.