Decisions in Thesaurus Construction and Use

Information Processing & Management, 43(4), 2007, 958-968.

Robert M. Losee
CB#3360
University of North Carolina
Chapel Hill, NC 27599-3360, USA
http://ils.unc.edu/~losee
email: *losee at unc period edu*

March 25, 2007

Abstract

A thesaurus and an ontology provide a set of structured terms, phrases, and metadata, often in a hierarchical arrangement, that may be used to index, search, and mine documents. We describe the decisions that should be made when including a term, deciding whether a term should be subdivided into its subclasses, or determining which of more than one set of possible subclasses should be used. Based on retrospective measurements or estimates of future performance when using thesaurus terms in document ordering, decisions are made so as to maximize performance. These decisions may be used in the automatic construction of a thesaurus. The evaluation of an existing thesaurus is described, consistent with the decision criteria developed here. These kinds of user-focused decision-theoretic techniques may be applied to other hierarchical applications, such as faceted classification systems used in information architecture or the use of hierarchical terms in "breadcrumb navigation."

1 Introduction

A thesaurus and an ontology provide a structuring to the concepts and terminology used by a discipline or that are found in a natural language. Thesauri provide lists of terms, often indicating structural relationships between the terms. An ontology provides what a thesaurus provides, as well as providing additional semantic and other information about the included concepts and relationships. The information a thesaurus or ontology provides is meant to be used ultimately both by indexers and indexing systems, as well as by searchers and end-users (Bates, 2002; Greenberg, 1997; Miller, 1997; Park & Sun Choi, 1996; Aitchison, Gilchrist, & Bawden, 2000). Thesaurus construction and use serve as fundamental functions within the fields of text mining and information retrieval.

The term or phrase entries in a thesaurus are commonly listed alphabetically for easy location of entries, with some entries being arranged hierarchically. Entries often indicate which other terms are *broader terms* (often abbreviated in a printed thesaurus as "BT") or *narrower terms* (often abbreviated as "NT"). Broader terms, often representing a *superclass*, such as mammals, are above narrower or *subclass* terms, such as *primates* or *ungulates*, on the hierarchy. Members of a subclass can be said to *inherit* features of the superclasses to which they belong. In the entry for *primates*, for example, one might find *humans* listed as a narrower term, while *mammals* might be listed as a broader term. Entries may also have *see* references, indicators that one should use the item one is encouraged to "see". For example, one might find an entry suggesting that instead of using the term *people* one should use the term *humans*, e.g., *people, see humans*. A *see also* reference, or related term reference (often abbreviated as "RT"), may indicate terms related to the same concept. Other informational notes may be provided in some entries.

In the work below, ways in which features included in thesauri benefit indexers and searchers are examined, including when features should or should not be included in a thesaurus. Precise knowledge about how people would use terms would simplify this problem. However, many decisions must be based upon estimates of our future behaviors or of the behavior of others, and thus the problem of deciding to include a term in a thesaurus may be viewed, in part, as an estimation problem.

The structure of a thesaurus may be hierarchical, and some of our examination of features and relationships between features is based on the notion of a concept hierarchy (Sowa, 2000). We may define a hierarchy as an ordering of sets or classes in which each set, or each item in a set, is immediately followed by zero, one, or more sets, and no set is followed eventually by a set that is also its predecessor; therefore, there are no cycles. Each set may be described as a class of entities or it may be the actual set of instances of the entities. For example, hierarchies are often used to represent a taxonomy of living creatures that suggests that animals might contain the sets of mammals and reptiles, mammals might contain primates, egg-laying mammals such as the platypus, and marsupials. The set of primates contains humans, gorillas, and so forth; this hierarchy may be produced manually or automatically, or produced consistent with precise criteria for class relationships (Sokal & Sneath, 1963). A hierarchical ordering may be found in a "faceted" classification system, in which a set may have as its immediate successor in the classification code either no set or a set whose nature has little to do with the nature of its predecessor, with the sets representing different facets or aspects of the class of objects being represented.

When developing a hierarchy of vocabulary or of concepts for retrieval or indexing purposes (Aitchison et al., 2000; Cleveland & Cleveland, 2001; Greenberg, 2001; Morita, Atlam, Fuketra, Tsuda, Oono, & Aoe, 2004), what should be the characteristics of the relationships between predecessors and successors? We refer here to a set in a hierarchy as the *parent* of one or more immediate successor sets, the latter refereed to as *child* sets. We suggest below a set of criteria by which one can judge whether a parent set should have one or more children sets; for example, should a broad term such as *human* be broken down into *female* and *male*? Criteria are also

provided for determining whether one type of class description of children is better than another type, such as whether *humans* are better defined as either *female* or *male* or whether humans should be subdivided into *adults* or *children*, or perhaps as those who like broccoli or those who dislike broccoli.

What criteria should one use when developing a thesaurus hierarchy? A hierarchy should be developed so that the performance when using the hierarchy is maximized. Below we will consider how performance is due to including or excluding thesaurus features, and thus how the design of a thesaurus may maximize performance.

2 Hierarchies and Features in Thesauri

Different thesauri, with their associated structures, may provide different qualities and quantities of support for users. Criteria for developing thesauri that support users' needs are developed below as decisions to be made based on expected user performance.

To address how terms in a thesaurus effect searching performance, we need to be specific about relevant feature characteristics and relationships. We denote a feature in hierarchy h by referring to the level in the hierarchy and the specific feature within that level. For different levels in h, h_1 through h_n, we may place different features on each level. The value of $h_{i,j}$ is feature j, or set of features j, at level i. Node j may itself be the root of a hierarchy. Thus, we might find a hierarchy within another hierarchy denoted as $h_{i,h_{j,k}}$. A hierarchy might represent *mammals* which consists of several classes, including *primates*, which, in turn, may be further subdivided into *apes*, *lemurs*, *monkeys*, and *humans*.

Classification systems using hierarchies provide representations which may then be used for clustering similar items together and for ordering the items, such as ordering books on a shelf arranged by the value of the volumes' Dewey Decimal Classification numbers. Classification systems often supply representations composed of components derived from a classification schedule or thesaurus. The quality of the underlying features in a thesaurus, determines the quality of the classification system, and thus the user's quality of browsing and retrieval from a classified collection.

One early theoretically supported classification system was suggested by Ranganathan. Ranganathan argues for the use of Personality, Matter, Energy, Space, and Time (PMEST) as the ordered classes for features or feature categories, while others such as Kaiser, Coates, and Vickery have proposed other conceptual categories for organizing information (Foskett, 1996). Given these categories, documents may be ordered based upon the Gray code, which can be shown to have certain optimal ordering properties for browsing (Losee, 1992, 2002).

Terms or phrases in a thesaurus must be chosen so that their use results in the best performance. The performance is usually library and community dependent. In a university library, using a term such as *kitten* would probably be inferior to using a term such as *cat*, whereas in a school library serving children below the

age of 10, terms such as *kitten* might frequently be used as search terms (Solomon, 1993). A graduate institution with a specialized biology library might find it most useful to use Latin genus-species identifiers that are familiar to most professional biologists. Terms may also refer to specific instances of the class, e.g., *Felix the Cat* or *Morris the Cat*, rather than to the class.

3 Performance of Ordering Systems

The Average Search Length (ASL) can be used as an empirical measure of document ordering performance (Losee, 2006b). As the average position of relevant documents in the ranked list of documents, ASL is easily interpreted. The value 1 would indicate that the average position of relevant documents is at location 1 (the first document in the ordered list) and, in an ordered list of N documents, an ASL value of N would indicate that the average position of relevant documents is at location N, the end of the ordered list of documents. In some circumstances, one may predict the ASL performance from the document set and relevance parameters, with the latter computed using available relevance feedback or attention metadata.

When the ASL (and related measures to be discussed below) use the complete set of all available documents, the measures and techniques discussed below may be used to develop a thesaurus or ontology most useful for *high recall* systems. When the ASL and related measures are computed using the first N documents, N is relatively small, and there are more than N documents in the set of all documents, the thesaurus that is developed is optimized for *high precision* systems, such as search engines that are best at retrieving a few good documents. By choosing an appropriate N, the thesaurus developer can produce a thesaurus optimized for any given position on the spectrum of systems ranging from *high precision* to *high recall*.

The Normalized Average Search Length ($\mathcal{A}$) represents the ASL scaled to the range of 0 to 1, with 0 being the upper bounds or best-case performance and 1 the lower bounds performance. At this scale, it may be interpreted as the percent of all documents that are ranked ahead of the average position of the relevant documents. If we double $\mathcal{A}$ we obtain the probability $\mathcal{W} = 2\mathcal{A}$; Assuming that the term is a positive discriminator, as is the case with most reasonable query terms, $\mathcal{W}$ will be a valid probability that represents the percent of documents in the first half of the ordered list that are ahead of the average position of relevant documents, or, expressed differently, the probability that a document in the first half is ahead of the average position of relevant documents. We assume that given system feature i with performance $\mathcal{W}_i$ and system feature j with performance $\mathcal{W}_j$, and features i and j are independent, the use of both features together to yield performance $\mathcal{W}_{i,j} = \mathcal{W}_i \times \mathcal{W}_j$.

The Relative Feature Utility, denoted as M, shows how many occurrences of independent identically distributed events of type j or a feature with the same discrimination power will yield the same performance as a single occurrence of feature

i (Losee, 2006b) and is computed as

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

For example, given values $\mathcal{A}_i = .3$ and $\mathcal{A}_j = .4$, we may compute $\mathcal{M} = \log(2 \times .3)/\log(2 \times .4) = 2.29$. Thus we can say that 2.29 occurrences of features of type j have the same discrimination power as a single occurrence of feature i or a feature of type i.

Using $\mathcal{M}$ to compare performance level x with a constant performance c and a second $\mathcal{M}$ comparing the upper bounds performance with the same constant performance c, one can compute the percent of the upper bounds performance found in performance at level x as $\mathcal{P}(x) = \log \mathcal{W}_x / \log \mathcal{W}_{Upperbounds_x}$. For example, if the $\mathcal{A}$ for the upper bounds is 0.1 and the performance of interest was $.4$, then the performance is computed as $\log(2 \times .4)/\log(2 \times .1) = 0.139$, meaning that this level of performance is 13.9% of the way from randomness toward the upper bounds performance level. When $\mathcal{A}_x = .1$ and the bounds are still at 0.1, we see that the performance is 100% of the way from randomness toward the upper bounds, and when $\mathcal{A}_x = .5$, the performance is at the 0% level.

4 Performance with Orthogonal Facets

Thesaurus performance will vary depending on whether one is searching using the terms in the thesaurus, assigning indexing terms from the thesaurus, or developing a thesaurus of possible index terms, or estimating the characteristics of the searching, indexing, or thesaurus development process. These performance measures may be computed *á posteriori*, after relevance judgments are made available, or the performance values may be estimated *á priori*, based upon estimates of the user's preferences and database characteristics. Estimates may also be used to predict the estimates of others, such as when a thesaurus developer tries to predict how indexers will use the thesaurus, and the indexer in turn is trying to predict the terms searchers will find useful.

Below, we use the subscripts T, I, and S to denote parameter and performance estimates made by the thesaurus developer, the indexer, or the searcher, respectively. We assume that estimates made by the searcher are of their own future performance. Estimates made by the indexer may be those of the searcher's expected performance. Estimates made by the thesaurus developer usually are those of the searcher's expected values if the indexer assigns this feature. Some estimates do not directly address the concerns of the searcher. When using a two letter subscript (from the three letters above), the first subscript denotes who is making the estimate and the second who the estimate is about, assuming that the second party may or may not be estimating the characteristics of the searcher. The subscript TI, for example, denotes a characteristic estimated by the thesaurus developer about the indexer, while TT represents the thesaurus developers estimates of his or her own interests. When only a single subscript is used, this implies that the estimation

is about the searcher. Thus, performance achieved or estimated when addressing the development of a thesaurus is $\mathcal{P}_T = \mathcal{P}_{TS}$.

If feature h_i has n possible values $h_{i,1}, h_{i,2}, \ldots, h_{i,n}$ the performance associated with feature $h_{i,j}$ may be denoted as $\mathcal{P}(h_{i,j})$. When using the entire set of features at level i, h_i, the performance may be denoted as $\mathcal{P}(h_i)$. We may estimate the performance of facet h_i in hierarchy h at level i, taken as a whole, as the average performance of the set of values for hierarchy level h_i, denoted as $E_i(\mathcal{P}(h_i, \mathcal{A}))$, with the expected value computed over all the values at level i. A searcher may use a single facet feature $h_{i,j}$, resulting in performance $\mathcal{P}_\mathsf{S}(h_{i,j}, \mathcal{A})$. When the average is computed over features at level i, one is computing the average based on the probability of use of each feature in this context, thus, this is a user centered performance measure. Because relevance judgments are used in computing $\mathcal{A}$, the $\mathcal{P}_\mathsf{S}(h_{i,j}, \mathcal{A})$ measure is user-specific.

The searcher may also estimate $\mathcal{A}$ as $\hat{\mathcal{A}}_\mathsf{S}$ based upon information available to the searcher. Parameters, those underlying characteristics of physical and social phenomena, may be estimated using a number of techniques (Fisher, 1925; Mosteller, 1968). Traditional estimation techniques, such as the method of moments or maximum likelihood estimation are widely used to produce quantitative estimates of means. An individual who estimates the characteristics of another individual or of a group needs to learn to initially estimate, and then calibrate their estimates (Cooper, 1978; Hey, 1983; Wagenaar & Keren, 1985; Suantak, Bolger, & Ferrell, 1996).

We may also estimate $\mathcal{A}$ for a single feature from its parameters. Assume $\mathcal{A} = (1 - p + t)/2$ in the case of a perfect ranking algorithm, where t represents the probability that a document has the feature in question and p is the probability that a relevant document has the feature (Losee, 1988, 2006b). We may compute t directly from the percent of documents having the feature, which is the same for all users of the system. The searcher may estimate p, denoted as $\hat{p}_\mathsf{S}$, from a mental experiment, from professional expertise, or from extrapolating from existing statistical data (Cooper, 1978). Given a $\hat{p}$ value, one can estimate the $\mathcal{A}$ and then the $\mathcal{W}$ and $\mathcal{P}$ values. We denote the searcher's estimate of the performance, given the searcher's estimate of p, as $\mathcal{P}_\mathsf{S}(h_{i,j}, \hat{p}_\mathsf{S}, t)$.

When beginning to construct a hierarchical level on a thesaurus or ontology, selecting a term with which to begin is a complex task. For many word senses, there exist a set of terms with similar meanings, synonyms, and a set of terms that occur together, referred to as related terms. Clearly we should select as the term to use, from the set of synonyms, that term which maximizes overall performance over all searches. This is a *global selection* rule. A simpler rule, known as the *local selection* rule, selects that term that has the best local performance from among queries consisting of just those terms in the synonym set. Synonyms and related terms may be automatically determined through statistical techniques that are sensitive to whether terms occur in similar contexts, that is, they have the same statistical relationships to other terms.

Using the local selection rule has drawbacks that can be addressed, if desired, by other selection algorithms. For example, terms that may have the best local performance from among a set of synonyms might themselves have a high ambiguity

measure (Losee, 2001) outside the set of synonyms, and thus would decrease overall performance. Those terms to be included in the list of possible synonyms should include only those terms with low levels of word sense ambiguity.

The performance associated with indexing a document by a term may be estimated from the performance values expected across the range of likely searchers. Estimation may be based upon an estimate of p by the indexer, $\hat{p}_{\mathsf{I}}$, with the resulting performance $\mathcal{P}_{\mathsf{I}}(h_{i,j}, \hat{p}_{\mathsf{I}}, t)$. The indexer also may directly estimate $\hat{\mathcal{A}}_{\mathsf{I}}$ and use $\mathcal{P}_{\mathsf{I}}(h_{i,j}, \hat{\mathcal{A}}_{\mathsf{I}})$.

When different searchers will have different p values for the term in question, the performance is estimated as

$$\mathcal{P}_{\mathsf{I}}(h_{i,j}, \vec{\hat{p}}_{\mathsf{I}}, \vec{t}) = \sum_{k \in searches} \Pr(k) \mathcal{P}_{\mathsf{S}}(h_{i,j}, \hat{p}_{\mathsf{I},k}, t_k). \quad (1)$$

Here, $\vec{\hat{p}}_{\mathsf{I}}$ represents the vector of different $\hat{p}_{\mathsf{I}}$ values, the list of estimates by the indexer(s).

The thesaurus developer's parameter estimates may take four different paths. A thesaurus developer may (1) make a thesaurus solely for their own purposes, ultimately disregarding how it might be used, (2) chose to estimate the indexers' use of the feature, (3) estimate the indexer's estimate of the searchers' parameters, or (4) estimate the searchers' parameters directly. The performance associated with using a feature in a thesaurus may be estimated from the performance values expected across the range of likely indexers and likely searchers. Estimation may be based upon an estimate of p by the thesaurus developer, $\hat{p}_{\mathsf{T}}$, with performance for searchers for feature j estimated as $\mathcal{P}_{\mathsf{T}}(h_{i,j}, \hat{p}_{\mathsf{T},j}, t_j)$ or one may directly estimate $\hat{\mathcal{A}}_{\mathsf{T},j}$ and the thesaurus developer may compute the performance $\mathcal{P}_{\mathsf{T}}(h_{i,j}, \hat{\mathcal{A}}_{\mathsf{T},j})$. We may denote the thesaurus developer's estimate of the indexer's parameters (without estimating the searcher's parameters) as $\mathcal{P}_{\mathsf{TI}}(h_{i,j}, \hat{p}_{\mathsf{TI},j}, t_j)$ or one may directly estimate $\hat{\mathcal{A}}_{\mathsf{I},j}$ and use $\mathcal{P}_{\mathsf{TI}}(h_{i,j}, \hat{\mathcal{A}}_{\mathsf{T},j})$. The estimate of the indexers characteristics is made here so that all indexers are treated as being the same, although one can treat them differently, as in Equation 1, or one may use the expected value for indexers. The method used to estimate $\mathcal{P}_{\mathsf{TT}}()$ is similar to that used for estimating $\mathcal{P}_{\mathsf{TI}}()$.

5 Decisions: Criteria for Using a Single Feature

A user who choses to search using thesaurus feature x (and we will assume here to simplify the presentation that only a single feature is being used) has performance when using this term denoted as $\mathcal{P}(x)$. Searching with term y similarly results in $\mathcal{P}(y)$. Faced with the decision to either use this term or to use no term at all, the searcher should use term x if and only if $\mathcal{P}_{\mathsf{S}}(x) \geq \mathcal{P}_{\mathsf{S}}(\emptyset)$, where $\emptyset$ denotes using no term, or the empty set. Similarly, a searcher should use term x instead of any other term y if and only if $\mathcal{P}_{\mathsf{S}}(x) \geq \mathcal{P}_{\mathsf{S}}(y)$ for any $y \neq x$.

A document should be indexed by term x rather than no term if and only if $\mathcal{P}_{\mathsf{I}}(x) > \mathcal{P}_{\mathsf{I}}(\emptyset)$. If we wish to study whether x is preferred over other possible terms,

the criteria for indexing with term x becomes $\mathcal{P}_{\mathsf{I}}(x) > \mathcal{P}_{\mathsf{I}}(y)$ for all y such that $y \neq x$.

The thesaurus should include the term x rather than no term at all if and only if $\mathcal{P}_{\mathsf{T}}(x) > \mathcal{P}_{\mathsf{T}}(\emptyset)$. If we wish to study whether x is preferred over other possible terms, the criteria for indexing with term x becomes $\mathcal{P}_{\mathsf{T}}(x) > \mathcal{P}_{\mathsf{T}}(y)$ for all y such that $y \neq x$.

In the situation where the system has two statistically independent facets, which can be viewed as placed in a hierarchical arrangement, one can multiply or combine the $\mathcal{W}$ values to yield $\mathcal{W}_{Parent,Child} = \mathcal{W}_{parent} \times \mathcal{W}_{Child}$. The $\mathcal{W}$ values are probabilities, denoting the chance that a document in the first half of the ordered list of documents is ahead of the average position of relevant documents, and if two $\mathcal{W}$ values are statistically independent, they may be multiplied to produce the joint probability. We may compute the performance value based on this $\mathcal{W}_{Parent,Child}$ value to produce $\mathcal{P}(h_{Parent}, h_{Child})$. Note that in a faceted classification system the features of interest will likely not be adjacent in the hierarchy; the same methods developed above apply to these non-adjacent features.

More generally, given n levels of a hierarchy with each level being independent, such as in a faceted classification system, the $\mathcal{W}$ value for the system for a producer or searcher is computed as $\prod_{i=1}^{n} \mathcal{W}_i$. The performance computed using $\mathcal{W}$ is $\mathcal{P}(h_1, h_2, h_3, \ldots h_{n-1}, h_n)$.

Using these estimates for calculating the performance with multiple facets, we may apply the techniques and make the decisions as described above.

6 Performance and Decisions when Using a Subclass

Many taxonomies include superclass-subclass relationships in which one class, such as *women*, is a full member of a superclass, such as *humans*. Given this relationship, we can assume that if one is a member of the subclass *women* then one is by necessity a member of the superclass of *humans*. This is similar to the notions of *broader terms* and *narrower terms* found historically in the literature about thesauri (Cleveland & Cleveland, 2001; Foskett, 1996).

When is it beneficial to subdivide a class into a superclass-subclass relationship? To answer this, one must be able to compute the performance of the two options so that one can then compare the expected performance of each option to determine which is preferable.

For this case, we may denote the performance associated with using the superclass *human* as $\mathcal{P}(human)$ and the performance associated with using the subclass *female* as $\mathcal{P}(female)$. We treat the subclasses of a superclass as mutually exclusive, so that we can subdivide *humans* into either *males* or *females*, but a human cannot be both *male* and *female*. Note that when the subclasses are allowed, e.g, *males* and *females*, the user may still choose to use the superclass, e.g., *humans*. One should

include just the superclass, instead of subdividing, only when

$$\sum_{i=v,u} \Pr(i)\mathcal{P}_{\mathsf{I}}(i)$$
$$\geq \Pr(h=n) \sum_{j=v,u} \Pr(j)\mathcal{P}_{\mathsf{I}}(j) + \Pr(h=n+1) \sum_{k=m,f} Pr(k)\mathcal{P}_{\mathsf{I}}(k), \quad (2)$$

that is, the left hand side, representing the superclass only, is greater than the right hand side, which includes both the superclass performance (and its probability) and the subclass performance (and its probability). We use m to denote males, f denotes females, u denotes humans and v denotes all the non-humans on the same hierarchical level as the humans. Here n represents the hierarchy level of the superclass ($h = n$)and $n+1$ the subclass hierarchy level ($h = n+1$). The probabilities for the left hand side and the right hand side may be different, even though notationally equivalent; values on the left hand side may be thought of as having a subscript indicating *left* and those on the right having a subscript indicating *right*. The probability of being at a given level on the right hand side is used to capture that users may look at one level, at a given set of terms, or they may choose a subclass level. For example, one might choose to search through a set of *countries*, or one might examine a subclass level of *linguistic dialects*. The left hand side of the inequality is when one may only consider the *country*, while the right hand side of the inequality shows the chance of being in the hierarchy at level n (*country*) and the chance of being at hierarchy level $n+1$ (*linguistic dialects*), yielding the expected performance computed over the two hierarchical levels on the right hand side of this inequality. When this inequality is true, performance including just the superclass term is superior to performance including the subclass terms, and when the equation is false, performance including the subclass terms is superior to performance including just the superclass term.

One can view these decisions graphically by examining when the left hand side of Equation 2 (represented in Figure 1 by the surface with no mesh) is greater than the right hand side of Equation 2 (represented in Figure 1 by the surface with mesh). In this example, we hold the following variables in Equation 2 constant; on the left hand side: $\Pr(u) = .5$, $\mathcal{P}(u) = .5$, $\Pr(v) = .5$, $\mathcal{P}(v) = .6$, and for the right hand side: $\Pr(u) = .5$, $\mathcal{P}(u) = .5$, $\Pr(v) = .5$, $\mathcal{P}(v) = .6$, $\Pr(h) = .9$, $\Pr(h+1) = .1$, and $\mathcal{P}(k_{male}) = .2$. On the left hand side of Figure 1 we can see where the unmeshed performance surface representing the use of the superclass only has performance superior to the meshed performance surface, representing the inclusion of the subclass, when the expected performance associated with including the feature *female* is lowest. The performance associated with the inclusion of the subclass is superior to that of using the superclass when the expected performance associated with the feature *female* is higher (the majority of the Figure).

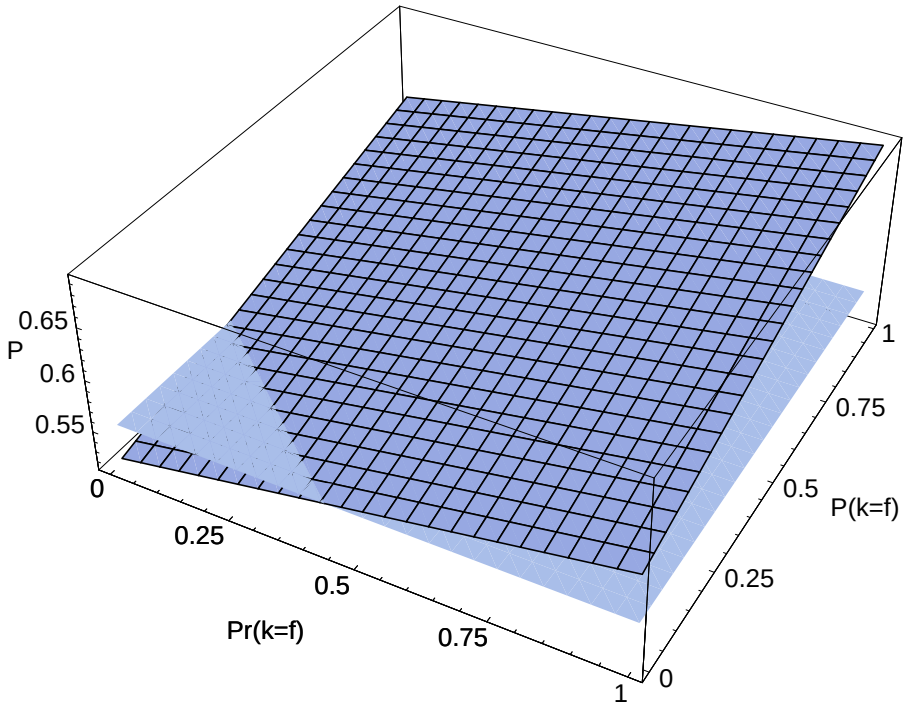

Figure 1: For a fixed set of parameter values and some varying values, the use of only a superclass is superior to that of using a superclass and a subclass (e.g., using *people* as well as the subclass *female* and *male*) when the unmeshed surface is above the mesh surface on the far left. The unmeshed surface represents the left hand side of Equation 2 and the meshed surface represents the right hand side of Equation 2.

7 Decisions Regarding Alternate Sets of Subclasses

In many cases, there are several alternatives available as subclasses of a superclass. Consider the situation where we have alternative subclasses, such as when members of a family may be divided into males or females, adults or children, etc. We will denote the set of n different members of subclass a as $a_1, a_2, \ldots, a_n$, with similar notation for subclass b. We may then express our problem as the question; should one use subclass a or subclass b?

The expected performance achieved when using only superclass s is denoted as $E\mathcal{P}(s)$ and the expected performance achieved when using the superclass s as well as subclass a is denoted as $E\mathcal{P}(s|a)$. The performance obtained when incorporating the subclass a is

$$\Pr(h = s|a)E(\mathcal{P}(s|a)) + \Pr(h = a) \sum_{i=a_1,a_2,\ldots,a_n} \Pr(i)\mathcal{P}(i),$$

the expected performance for the superclass given subclass a combined so that the probability the user will choose the superclass weights the performance at that level. The probability the user will choose the subclass level in the hierarchy is used to weight the expected performance at that level.

The thesaurus developer should choose to use subset a instead of subset b if and only if

$$\begin{aligned} &\Pr(h = s|a)E(\mathcal{P}_{\mathsf{T}}(s|a)) + \Pr(h = a) \sum_{i=a_1,a_2,\ldots,a_n} \Pr(i)\mathcal{P}_{\mathsf{T}}(i) \\ &\quad \geq \Pr(h = s|b)E(\mathcal{P}_{\mathsf{T}}(s|b)) + \Pr(h = b) \sum_{j=b_1,b_2,\ldots,b_n} \Pr(j)\mathcal{P}_{\mathsf{T}}(j). \end{aligned} \tag{3}$$

The decisions that would be made using the decision rule provided by Equation 3 may be seen in the example data shown in Figure 2. This illustrates when performance using subclass a, represented by the unmeshed surface, will be superior to performance using subclass b, represented by the meshed surface. For illustrative purposes here we assume the following parameter values: $\Pr(h = s|a) = .4$, $\Pr(h = s|b) = .41$, $E\mathcal{P}_{\mathsf{T}}(s|a) = .65$, $E\mathcal{P}_{\mathsf{T}}(s|b) = .6$, and the sum on the right hand side is set to .6. EP() in Figure 2 represents the expected performance sum on the left hand side of Equation 3. We can see on the upper right hand side of Figure 2 that as the left hand side of Equation 3 increases, as represented by the plain performance surface, the performance of the subclasses represented by the left hand side of Equation 3 and using the associated subclass (a) exceeds that of the right hand side and the associated subclass (set b).

8 Procedure for Developing a Thesaurus

The automated construction of a thesaurus has been motivated in past studies by linguistic and retrieval concerns (Fox, Nutter, Ahlswede, Evens, & Markowitz, 1988;

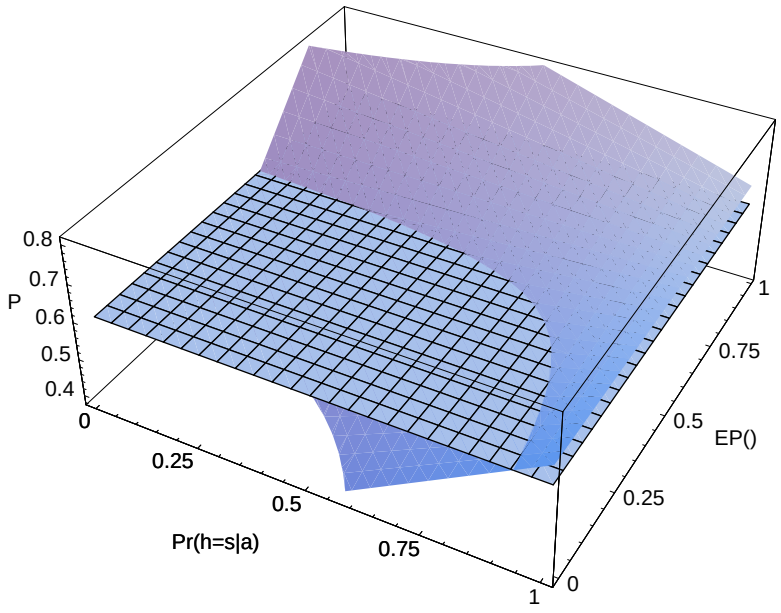

Figure 2: Using one subclass is better than another subclass when one performance surface is above the other. This might be used to show when dividing *people* into *females* and *males* (the a set in Equation 3) is better than dividing *people* into *adults* and *children* (the b set in Equation 3). The plain surface represents the left hand side of Equation 3 (*females* and *males*) and the meshed surface represents the right hand side (*adults* and *children*). EP() in the Figure represents the expected performance sum on the left hand side of the Equation.

Table 1: Three different example thesauri. Upper bounds are assumed to be 0.01.

Thesaurus	*Term*	*Type*	$\mathcal{A}$	$\Pr(term)$
1	human	class	0.4	1.0
2	human	superclass	0.2	0.6
	female	subclass	0.3	0.2
	male	subclass	0.4	0.2
3	human	superclass	0.123	0.9
	adult	subclass	0.07	0.05
	child	subclass	0.07	0.05

Park & Sun Choi, 1996; Miller, 1997). Given the decision criteria developed above for use by thesaurus developers, indexers, and information searchers, we may develop a thesaurus based on user needs.

A thesaurus should be developed so that one maximizes $\mathcal{P}_{\mathsf{T}}(thesaurus)$, the performance of the entire thesaurus. One can begin to develop a thesaurus by first determining which feature i has the highest $\mathcal{P}_{\mathsf{T}}(i)$ and has the highest performance when compared to its synonyms and related terms. We place this at level one in the hierarchy; the term may be denoted as $h_{1,1}$. We then select those terms that should go below this term in the hierarchy, using the criteria above. The highest performing remaining term is then placed at level h_1 as a new feature, and subclasses may then be added to this term. Related terms may be added to terms if desired by the thesaurus developer. This process is repeated until all terms are included above the noise level. To minimize noise in thesaurus construction, it may be desirable to set a constant, such as 3%, by which the left hand side of the above equations must exceed the right hand side.

The data needed to make these decisions are assembled into performance measures using the techniques shown above, which provides sample calculations. These techniques have also been discussed elsewhere (Losee, 2006b). Using these performance values, one may use Equation 2 or 3 to determine which structure would be best to use. Figures 1 and 2 show how ranges of data may be applied to practical situations, illustrating the relationships in these decision rules.

9 Evaluating a Thesaurus and Its Components

These techniques may be used to evaluate an existing thesaurus. Performance data needs to be accumulated from searches and relevance judgments, or their surrogates. Consider the fictitious data in Table 1 for 3 different and independent thesauri. The performance for thesaurus 1, 2, and 3 is 6%, 18%, and 37%, respectively.

Using the criteria in Equation 2, Thesaurus 1 is inferior to Thesaurus 2 and Thesaurus 1 is inferior to Thesaurus 3, suggesting that subdividing *humans* into one of these two types of subclasses is appropriate.

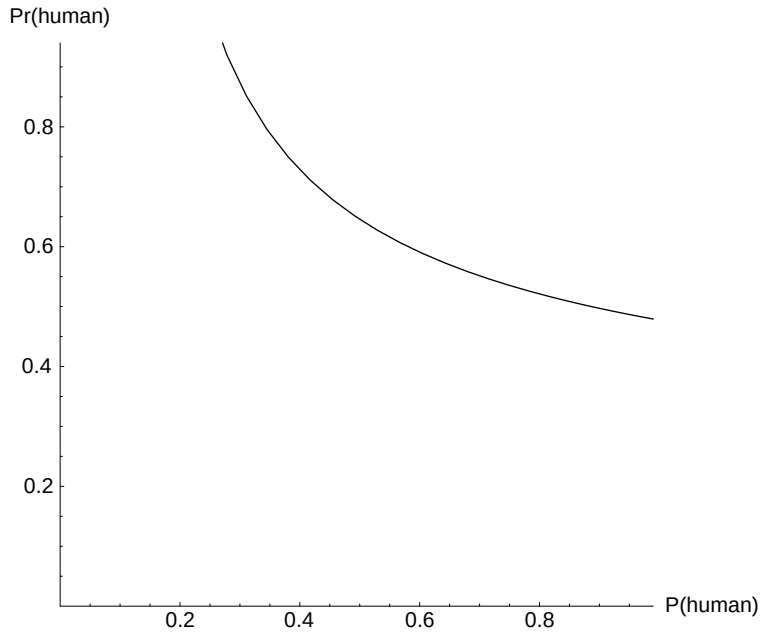

Figure 3: The break-even point for the *human* values in Thesaurus 3 in Table 1 that will result in Thesaurus 3 having the same level of performance as Thesaurus 2. The x axis represents values for $\mathcal{P}(human)$ and the y axis represents values for $\Pr(human)$.

Using the criteria in Equation 3, Thesaurus 2 is inferior to Thesaurus 3, suggesting that subdividing *humans* into *adults* and *children* is superior to dividing *humans* into *females* and *males*.

One can use data such as this to further explore the nature of this thesaurus and domain specific characteristics of its sublanguage. For example, if we wish to understand the relationship between the $\mathcal{A}$ and probability parameters in Thesaurus 3 above, we might algebraically solve to find those parameters that produce the same overall thesaurus performance as Thesaurus 2. Figure 3 shows the set of values for the *human* component of Thesaurus 3 that produces the same set of values for the thesaurus, as a whole, when compared to Thesaurus 2, taken as a whole. Note that as the probability $\Pr(human)$ varies, we correspondingly vary $\Pr(adult)$ so that the probabilities sum to 1. The $\mathcal{P}(adult)$ value remains fixed.

10 Discussion and Conclusion

The decisions one makes when incorporating subject terms into a thesaurus or ontology, or when using those terms to index or search, may be based on the expected performance associated with the decision options. Above we discussed how to incorporate a particular model of document ordering performance, and this is then used in developing criteria for various combinations of terms in a thesaurus. Because of the probabilistic nature of these performance measures, they may be estimated and thus we can discuss how a thesaurus developer, for example, may estimate the characteristics of a searcher. Criteria that are sufficient for development are also sufficient to measure performance, and we have provided an example of how to evaluate and compare different thesauri. Additionally, these techniques also may be applied to other hierarchical applications, such as faceted classification systems used in information architecture, or the use of hierarchical terms in "breadcrumb navigation."

The development of these rules has been based on the notion of expected performance using a set of features, and with the expectation computed over a set of uses, either present uses, expected future uses, or a combination of both. Using this notion, we can summarize the above rules as follows: Include a set of subclass members at hierarchy level h_{i+1} of a superclass at hierarchy level h_i when the expected performance of the subclass combined with the superclass is greater than the expected performance of the superclass alone. Which of two or more alternative feature sets should be used? From among a set of possible feature sets of size 1 to size n at a given level, choose the set (regardless of size) that has the largest expected performance.

The decision-based model proposed here explicitly captures the uncertainty when humans make decisions. There is uncertainty in the relationship between index terms and whether a user would use them if they found a document relevant. The indexer must estimate this uncertainty when choosing to assign or not assign index terms, adding a second level of uncertainty to that experienced by the user alone. A third level of uncertainty exists in the decision to include a term in a thesaurus or ontology.

Using these criteria, a *static thesaurus* may be evaluated or developed. One also can develop an adaptive or *dynamic thesaurus* that adapts to the individual or group, with each user set having its own needs-optimized thesaurus, vocabulary, and system. Different sets of users with semantically heterogeneous information sources may develop their own thesauri "from scratch" or they may begin with a "starter" or neutral thesaurus and then adapt it to better address the users' needs. The latter is far more efficient and has a much shorter learning curve. With existing techniques and technology, as well as the ability to dynamically order structured and unstructured information (Losee, 2006a), the use of a dynamic thesaurus to provide control of the concepts used to order documents will provide improved performance over the next decade of document and media systems.

References

Aitchison, J., Gilchrist, A., & Bawden, D. (2000). *Thesaurus Construction and Use: a Practical Manual* (Fourth Edition edition). Fitzroy Dearborn Publishers, Chicago, IL.

Bates, M. J. (2002). After the dot-bomb: Getting web information retrieval right this time. *First Monday*, *7*(7). http://firstmonday.dk/issues/issue7_7/bates/index.html.

Cleveland, D. B., & Cleveland, A. D. (2001). *Introduction to Indexing and Abstracting* (Third edition edition). Libraries Unlimited, Englewood, Colo.

Cooper, W. S. (1978). Indexing documents by Gedanken experimentation. *Journal of the American Society for Information Science*, *29*(3), 107–119.

Fisher, R. A. (1925). Theory of statistical estimation. *Proceedings of the Cambridge Philosophical Society*, *22*(5), 700–725.

Foskett, A. C. (1996). *The Subject Approach to Information* (Fifth edition). Library Association Pub., London.

Fox, E. A., Nutter, J. T., Ahlswede, T., Evens, M., & Markowitz, J. (1988). Building a large thesaurus for information retrieval. In *Proceedings of the Second Conference on Applied Natural Language Processing*, pp. 101–108 Morristown, NJ. Assocation for Computational Linguistics.

Greenberg, J. (1997). Reference structures: Stagnation, progress, and future challenges. *Information Technology and Libraries*, *16*(3), 108–119.

Greenberg, J. (2001). Optimal query expansion (QE) processing methods with semantically encoded structured thesauri terminology. *Journal of the American Society for Information Science and Technology*, *52*(6), 487–498.

Hey, J. D. (1983). *Data in Doubt*. Basil Blackwell, Oxford.

Losee, R. M. (1988). Parameter estimation for probabilistic document retrieval models. *Journal of the American Society for Information Science*, *39*(1), 8–16.

Losee, R. M. (1992). A Gray code based ordering for documents on shelves: Classification for browsing and retrieval. *Journal of the American Society for Information Science*, *43*(4), 312–322.

Losee, R. M. (2001). Natural language processing in support of decision-making: Phrases and part-of-speech tagging. *Information Processing and Management*, *37*(6), 769–787.

Losee, R. M. (2002). Optimal user-centered knowledge organization and classification systems: Using non-reflected gray codes. *Journal of Digital Information*, *2*(3). http://jodi.ecs.soton.ac.uk/Articles/v02/i03/Losee.

Losee, R. M. (2006a). Browsing mixed structured and unstructured documents. *Information Processing and Management*, *42*(2), 440–452.

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

Miller, U. (1997). Thesaurus construction: Problems and their roots. *Information Processing and Management*, *33*(4), 481–493.

Morita, K., Atlam, E., Fuketra, M., Tsuda, K., Oono, M., & Aoe, J. (2004). Word classification and hierarchy using co-occurrence word information. *Information Processing and Management*, *40*, 957–972.

Mosteller, F. (1968). Association and estimation in contingency tables. *Journal of the American Statistical Association*, *63*(321), 1–28.

Park, Y. C., & Sun Choi, K. (1996). Automatic thesaurus construction using Bayesian networks. *Information Processing and Management*, *32*(4), 543–553.

Sokal, R. R., & Sneath, P. H. A. (1963). *Principles of Numerical Taxonomy*. W. H. Freeman, San Francisco.

Solomon, P. (1993). Children's information retrieval behavior: A case analysis of an OPAC. *Journal of the American Society for Information Science and Technology*, *44*(5), 245–264.

Sowa, J. E. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole, Pacific Grove, CA.

Suantak, L., Bolger, F., & Ferrell, W. R. (1996). The hard-easy effect in subjective probability calibration. *Organizational Behavior and Human Decision Processes*, *67*(2), 201–221.

Wagenaar, W. A., & Keren, G. B. (1985). Calibration of probability assessments by professional blackjack dealers, statistical experts, and lay people. *Organizational Behavior and Human Decision Processes*, *36*, 406–416.