11 LINGUISTIC KNOWLEDGE

He multiplies words without knowledge.

—Job 35:16

11.1 INTRODUCTION

Natural language processing contributes to improved retrieval performance by extracting from natural language text information about terms and their relationships. This information is far richer than what is obtained with term frequency methods that assume the statistical independence of terms. Once acquired, this linguistic knowledge may be reflected in retrieval and filtering systems in either a modified query or in a modified document incorporating this information. *We can expect natural language processing to improve filtering and retrieval performance if, and only if, the application of linguistically derived information increases the ability of the retrieval system to discriminate between documents of differing relevance.* While linguistic knowledge may be obtained through purely statistical analysis, humans may extract this same information without using massive number crunching capabilities, and it is likely that, for automated systems, linguistic methods may be ultimately simpler and faster at extracting information that improves retrieval performance than are methods that explicitly incorporate higher order statistical dependencies. We present a model of grammatical parsing and part-of-speech tagging that allows us to make specific claims about the level of retrieval and filtering performance that will be obtained when linguistic knowledge is incorporated. The model provides both upper and lower bounds for performance with the best-case and worst-case part-of-speech tagging.

The purpose of this examination of how natural language processing improves retrieval and filtering performance is to

- determine expected performance,
- make precise statements about what complex of linguistic structures and applications result in superior filtering performance, and
- learn how individual linguistic factors affect performance.

We can accomplish these goals through the application of the analytic models discussed earlier. This chapter is an extension of Chapter 8, which addressed the performance of systems incorporating statistical relationships between terms.

Most research examining either linguistic phenomena or information retrieval and filterings systems, separately or in conjunction, describe system performance by examining average retrospective performance figures, or the characteristics of specific instances or phenomena. However, exact *expected* results may be obtained through modeling enough aspects of a filtering system to provide information about the expected characteristics of the system. This is different than computing a single average for an experimental data set. The analytic approach allows us to formally understand how linguistic phenomena interact to produce a certain level of performance. Analytic techniques complement experimental methods and empirical data-gathering methodologies. While the impact of some assumptions, as well as some models of natural language processing, may best be studied empirically, many other questions addressing the relationship between natural language processing and information filtering are best answered analytically.

The purpose of applying linguistic knowledge to documents and queries in filtering systems is to provide additional information about individual terms and sets of terms beyond the simple frequency of term occurrences. This occurs primarily through two mechanisms: part-of-speech tags (e.g. *noun*, *verb*, or *subject*), and structure producing phenomena, such as grammatical parsing, that can be used to identify relationships between terms. For example, the phrase representing the subject of the sentence, *The brown spot was on the green shirt,* refers to a brown spot. Identifying the spot as brown, as opposed to green, requires a parsing procedure which shows that the relationship between brown and spot is different than the relationship between spot and green and that *the green shirt's brown spot* has a different and what intuitively seems to be a stronger relationship between *green* and *shirt* than between *brown* and *shirt.*

To design and understand retrieval and filtering, we need to understand how the output of linguistic processes affect text filtering and retrieval performance through statistical dependencies between terms, as well as through the choice of terms and attached part-of-speech tags. Terms in natural language may carry information that may be elicited more easily and more accurately through linguistic rather than statistical techniques. Whether one approach or another is better in some cases, or in all cases, can be determined analytically, given accurate parameter values.

The basic question examined in this chapter is

> *under what circumstances will incorporating linguistic knowledge result in superior filtering performance, and under what circumstances will performance grow worse when linguistic techniques are applied?*

By providing analytically based statements of when a linguistic method improves retrieval performance, combined with the results of earlier chapters, we can answer

Table 11.1. Sample documents, where "Y" denotes *yes* and "N" denotes *no*.

Relevant Documents		*Non-relevant Documents*	
Term Present?	*Query Tag?*	*Term Present?*	*Query Tag?*
Y	Y	Y	N
Y	Y	Y	Y
Y	N	N	N
N	N	N	N
N	N	N	N

this question. If the ability to filter information in a retrieval or network application is reflective of what allows humans to filter information and retrieve facts, our results may be indicative of the utility of linguistic methods for humans in all domains, not just filtering and retrieval.

11.2 TAGGING AND SUFFIX STRIPPING WITH ONE TERM

Grammatical part-of-speech tags are assigned in a computational environment though the *tagging* process (Brill, 1994). Tagging may be performed by a tagging program that operates based on either grammatical rules that are programmed into the system or learned, or tags may be assigned based upon statistical considerations, often using Hidden Markov Models. Assigning a part-of-speech tag to a term token allows users to discriminate between different senses in which a term is used. A part-of-speech tag assigned to the term *girl* in sentences like *girl bites dog* and *dog bites girl* can be used by a retrieval system to retrieve only those documents in which a girl acts as biter (subject) or those documents in which the girl is bitten (object). While using natural language processing techniques such as tagging may improve retrieval and filtering performance, the degree of improvement with word-sense disambiguation may vary from small to moderate amounts (Burgin & Dillon, 1992; Krovetz & Croft, 1992; Sanderson, 1994; Strzalowski, 1995). The analytic rules described below can lead to an understanding of how tagging contributes toward the matching of queries and text.

We initially make some simplifying assumptions to allow us to understand the nature and benefits of part-of-speech tagging. For example, we limit ourselves to queries with a single term to allow us to examine the impact of tagging on a single term, taken in isolation. We thus treat a single (query) term and its tag together as a single complex. We also assume optimal retrieval so that we can avoid the complexity added by working with a sub-optimal ranking method. Earlier work on analytic prediction of retrieval performance shows how a model may assume multiple terms (Chapter 8) and sub-optimal retrieval (Chapter 5), but such complex models may conceal some simple underlying phenomena.

The basic unit of linguistic analysis is the *term* or word. We refer to an occurrence of a term as a term *token* and the generic form for the term as the term *type*. The distinction is roughly that between numeral and number (the former corresponding to token, the latter to type).

What constitutes a term is open to debate; we follow the convention that text strings that are bounded by spaces are terms. This crude definition is adequate in most circumstances, although questions are not addressed such as whether more complex expressions such as *take-it-or-leave-it*, *Greco-Roman*, or *computer-human interface* are best treated as one term or two, or whether each phrase is a single or multiple concept. Phrases contain groups of terms that together address a particular concept or construct. Sentences are then composed of one or more phrases representing an entire statement. Parsing decomposes a sentence into phrases and individual terms occurrences, each with part-of-speech tags attached. The output of a parser, a *parse tree,* is a hierarchy of tags assigned to a sentence, with tags being assigned to individual terms as well as to phrases. Phrases and entire sentences are discussed in later sections of this chapter.

Part-of-Speech Tags and Single Term Performance

Understanding how part-of-speech tags can affect retrieval performance can lead one to make better decisions about the use of natural language processing in filtering and retrieval systems. When tagging query terms in documents, the number of documents with the tagged query term will be the same as or fewer than would be the number of documents with the untagged term if no tagging were used. As before, the probability that the query term occurs in a document is denoted as t, and the probability that a document has the query term, given that the document is relevant, is denoted as p. The probability that a document is tagged with the query tag, given that it has the query term, is denoted as τ. The probability that a document has the query term and is tagged with the query tag is the product $t\tau$. Similarly, the probability that a term is tagged with the query tag, given that the document has the term and is relevant, is π. The probability that a relevant document contains the term tagged with the query term's tag is the product $p\pi$.

Theorem 11.1 (Tagging improving performance) *Given optimal ranking, performance is improved if and only if*

$$1 + t\tau - p\pi < 1 + t - p. \tag{11.1}$$

Here the left hand side represents $\mathcal{A}$ with tagging and the right hand side $\mathcal{A}$ without tagging, where $\mathcal{A}$ (Equation 7.1) is the expected position of a relevant document in the ordered list of documents, scaled to be in a range from 0 to 1. A term is assigned a part-of-speech tag with the expectation that the tagging will increase retrieval or filtering performance. Filtering performance is improved with tagging if and only if the ASL with the tagging is less than the ASL without the tagging. This condition is met when the $\mathcal{A}$ with tagging is less than the $\mathcal{A}$ value without tagging.

Corollary 11.1 (Tagging decreasing performance) *Tagging decreases retrieval performance when*

$$1 + t\tau - p\pi > 1 + t - p.$$

We can measure the degree to which retrieval performance with tagging exceeds the performance without tagging by reformulating Equation 11.1 so that a Tagging Improvement Factor (TIF) represents the amount added to the left hand side of

Equation 11.1 by tagging, with a positive value for the TIF indicating that tagging improves performance (decreases ASL), and a negative value indicates that tagging decreases system performance.

Theorem 11.2 (Tagging improvement factor) *The* Tagging Improvement Factor *(TIF), the improvement in $\mathcal{A}$ when part-of-speech tagging is used, is computed as* $TIF = t(1-\tau) - p(1-\pi)$.

This follows from

$$1 + t\tau - p\pi + \text{TIF} = 1 + t - p \tag{11.2}$$

and then

$$\text{TIF} = t(1-\tau) - p(1-\pi). \tag{11.3}$$

The TIF is thus computed as the proportion of all documents with the term in question that aren't tagged minus the proportion of relevant documents with the term in question that aren't tagged. A TIF is 0 when the same proportion of all documents as relevant documents are assigned the tag in question. This occurs when the tags are distributed the same in both the set of relevant documents and in all documents. If a tag occurs with greater relative frequency in the relevant documents, the tagging results in improved performance.

Corollary 11.2 (Change in ASL due to tagging) *The change in ASL for a single term due to part-of-speech tagging may be computed as*

$$ASL_{\Delta} = N\left(\mathcal{Q}(\mathcal{A} - TIF) + \overline{\mathcal{Q}}(\overline{\mathcal{A}} + TIF)\right) \tag{11.4}$$

where ASL is the expected position of relevant document (Equation 7.5) and $\mathcal{Q}$ is the probability the ranking used is optimal (Equation 6.1).

Example. The data in Figures 11.1 and 11.2 show the break-even points between text filtering performance with and without part-of-speech tags, that is, where the TIF is 0. Giving varying values of p, τ, and π, and $t = 1/10$ for Figure 11.1 and $t = 1/2$ for Figure 11.2, we find that, for lower values of t, the p value rises to both higher values and rises more quickly than for higher values of t. The area representing positive TIF values (improved performance with tagging) is above the surface, with area below the surface representing values that will decrease performance. The range of values that will result in improvements with tagging increases with t.

When tagging is applied to the data in Table 11.1, where $\tau = 1/2$ and $\pi = 2/3$, we compute the ASL as $\frac{10}{2}(1 + \frac{1}{2}\frac{1}{2} - \frac{2}{3}\frac{3}{5}) + \frac{1}{2} = 4.75$. If one examines Figure 11.2 one finds that for $t = 1/2$, $p = 3/5$, $\pi = 2/3$ and $\tau = 1/2$, we are at a point above the break-even surface but not very far from the surface, indicating that tagging helps somewhat in this case.

Figure 11.3 shows the ASL when $t = 1/2$ and $p = 3/5$ for both a tagged query (the fine mesh) and for an untagged query (with larger holes in the mesh). Figure 11.4 similarly shows the ASL when $t = 1/10$, with everything else similar except for the ASL. For the case where there is a low t and a large gap between t and p, the terms are already good discriminators and the region where tagging will improve performance (the top part of the figure where the large mesh is above the finer mesh, the latter

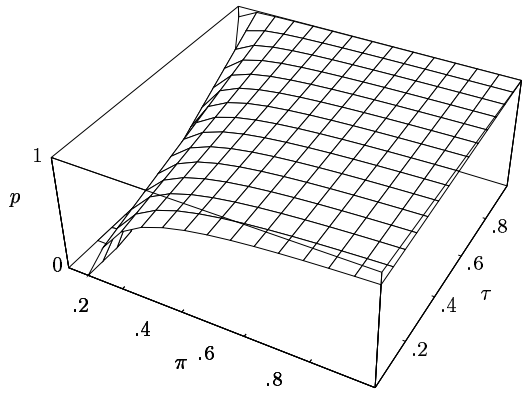

Figure 11.1. The break-even π points for deciding to tag or not tag a term, with $t = 1/10$.

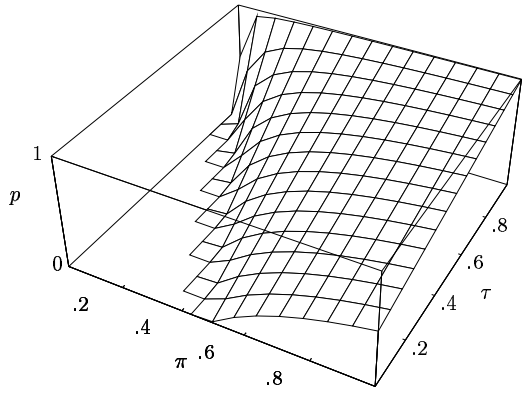

Figure 11.2. The break-even π points for deciding to tag or not tag a term, with $t = 1/2$.

representing tagged performance) is small. For this data, tagging results in improved performance with very high π and, to a lesser extent, for lower τ.

Best and Worst-case Performance with Tagging

Retrieval performance may be viewed as being dominated by t and p. Part-of-speech tagging may be viewed as providing a means of improving on this by modifying these values through τ and π. However, part-of-speech tagging cannot overcome some limits imposed by the untagged probabilities. Note that our initial development assumes very large databases and associated approximations; later, some exact values will be computed which will be accurate with small databases.

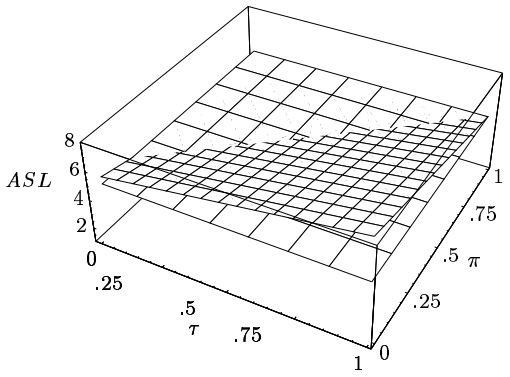

Figure 11.3. The Average Search Lengths (ASL) for two searches of ten sentences. The large mesh (large holes) represents retrieval with no tags and with $t = 1/2$ and $p = 3/5$. The smaller mesh represents a search with tagging, where the parameters are as above but where π and τ are allowed to vary.

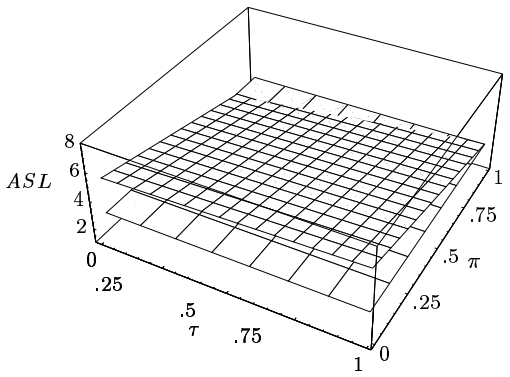

Figure 11.4. The Average Search Lengths (ASL) for two searches of ten documents. The large mesh (large holes) represents retrieval with no tags and with $t = 1/10$ and $p = 3/5$. The smaller mesh represents a search with tagging, where the parameters are as above but where π and τ are allowed to vary.

Theorem 11.3 (Approximate best-case tagging performance) *The best performance obtained with tags is*

$$ASL = \frac{N}{2}(1-p) + \frac{1}{2} \tag{11.5}$$

and the corresponding best-case $\mathcal{A}$ is $1 - p$.

The highest TIF (Equation 11.3) occurs when the proportion of all documents with the query term tagged as in the query approaches its minimum ($\tau \to 0$) and the proportion of relevant documents with the query term tagged as in the query approaches its maximum ($\pi \to 1$). This approaches,

$$ASL = \frac{N}{2}(1 + 0t - 1p) + 1/2 = \frac{N}{2}(1 - p) + \frac{1}{2}.$$

Corollary 11.3 (Approximate worst-case tagging) *The worst-case performance obtained with tags is*

$$ASL = \frac{N}{2}(1 + t) + \frac{1}{2} \tag{11.6}$$

and the corresponding worst-case $\mathcal{A}$ is $1 + t$.

The worst case occurs when $\tau \to 1$ and $\pi \to 0$. In this case,

$$ASL = \frac{N}{2}(1 + 1t - 0p) + 1/2 = \frac{N}{2}(1 + t) + \frac{1}{2}.$$

We can see that performance will be proportionally no better than would be obtained with an $\mathcal{A}$ factor of $(1-p)/2$ and no worse than $(1+t)/2$. The range of retrieval performance is thus bracketed between that obtainable with optimal tagging and that with the worst case tagging.

In reality, τ doesn't approach very close to 0 with small databases, such as in Table 11.1, although τ does approach very close to 0 with larger databases. With smaller databases, or when a close approximation isn't satisfactory for a large database and it is desirable to compute the exact value, it becomes necessary to take into account how close τ is to 0 when computing the best-case and worst-case performance values. This is because the limits described earlier are not met (by a factor gp, where generality $g = \Pr(rel)$).

Theorem 11.4 (Exact best-case tagging performance) *The exact upper performance bounds with part-of-speech tagging and generality g are*

$$ASL = \frac{N}{2}(1 + gp - p) + \frac{1}{2}. \tag{11.7}$$

This assumes that $\pi = 1$ and $\tau \to 1$ as in Equation 11.5. There must be at least (gp) tagged terms in the document database. The gp values in Equation 11.7 becomes very small, approaching 0 when a very large, realistic database is used, so that this equation in the limiting case approaches Equation 11.5.

In most realistic searches, t will be rather small if it is a good search term, usually below .01. When p is much higher than this, as would be the case with a strongly discriminating term, the potential for improvement with tagging is far better than the potential for a decrease in performance.

Using the data in Table 11.1, $N = 10$, $p = 3/5$, $t = 1/2$, and $g = 1/2$, and using the approximate best and worst case measures described in Equations 11.5, and 11.6, $8 \geq ASL \geq 2.5$. Using the exact formula (Equation 11.7) however, we find

the $ASL \geq 4$. The ASL computed earlier for this tagged query was 4.75. It is clear that this is better than the ASL of 5 found when no tagging occurs but is not as good as it could be, not equaling the best-case performance (4). Note that random retrieval here would result in $ASL = 5.5$.

Construction of Optimal Tags

Continuing with this example, optimal or best-case tagging and the ASL of 4 could be obtained here by tagging all relevant documents with the term in Table 11.1 and by not tagging those non-relevant documents with the term. In this situation, the three relevant documents with the term would be at the beginning of the ranked list (average position 2) and the other two relevant documents would be at the average position of 7, producing $ASL = 4$.

This suggests the following:

Theorem 11.5 (Constructing optimal tags) *Given that the upper bounds of performance are obtained as with Equation 11.7, where the active component of $\mathcal{A}$ is $1 + \tau t - \pi p$ and where $\tau \to 0$ and $\pi \to 1$, the best-possible tagging may be constructed by tagging all relevant documents containing the term with the query tag and none of the non-relevant documents with the term with the query tag.*

Clearly, this has the effect of maximizing π and minimizing τ, subject to the constraint imposed by g which is assumed to be fixed above. We see that there must be at least gp documents with the tagged term; if these are all relevant, then the upperbounds are obtained.

Stemming and Suffix Removal

Stemming terms, that is, removing suffixes from terms, may be seen as a form of "untagging" of terms, finding a commonality between this term and another term by removing information that previously made a term unique. Stemming may improve performance (Harman, 1991; Popovic & Willett, 1992; Porter, 1980) by increasing recall (Kraaij & Pohlmann, 1996). Stemming terms such as *stemming* can produce a stem such as *stemm* or *stem*. In a sense, it is not important whether the stem is that which would be produced by a human; what is important is that the stems be consistent, so that if one were looking for works on *stemming*, one would find them.

Terms with stems correlate weakly with the stemmed terms, suggesting that there is a relationship between the root and the unstemmed version (Church, 1995). This correlation may approach one half for terms that are good discriminators and approaches zero for terms that are poor discriminators.

Different stemming methods yield different levels of performance, with the relative performance levels depending significantly on the choice of performance measure used; which is the best stemmer procedure for a given situation is still an open question (Paice, 1994).

Using the inverse of the methods described above, the performance of systems using stemming in a single term case may be studied. Modifying Equation 11.2, we may say that the $\mathcal{A}$ factor without stemming (with suffixes and thus similar to tagging with part-of-speech tags) is $1 + \tau t - \pi p + c$, while, with stemming (untagging is similar to suffix removal), the A factor is $1 + t - p$, where c is the Suffix Decreasing Factor (SDF), the decrease in performance due to c set as in Equation 11.3. The best

and worst case performance figures without stemming (with tagging) are as in earlier discussions about part-of-speech tagging; their development is posed as an exercise below.

EXERCISE 11.1. [*Easy*] Assume that there are 100 documents, half with the term, and 50 of the documents are relevant, 40 of them with the term. When considering the tagging of terms, the term (with the query tag) is in 40 of the documents and in 35 of the relevant documents. What is $\mathcal{A}$ with and without tagging? What is the ASL with and without tagging? Does part-of-speech tagging help or hurt performance in this case?

EXERCISE 11.2. [*Moderate*] What are the exact lower bounds for retrieval performance with tags (the complement to Equation 11.7)?

EXERCISE 11.3. [*Difficult*] Develop a full set of theorems and conjectures for stemming that are similar to those developed for part-of-speech tagging.

11.3 MULTIPLE TAGS PER TERM TYPE

Filtering performance can obviously be enhanced through the use of supplemental part-of-speech tags. It is also clear that tagging can decrease performance. While the preceding section addressed the use of single tags with single term queries, it is helpful to consider performance when a single term in an untagged environment can explicitly produce two different tagged terms, one tagged as with the query and the other tagged differently. For example, the untagged term *run* may be labeled as either a verb, such as would occur in statements such as *Fred will run for president* or *Caitlyn will run after the cat,* while *run* labeled as a noun would be found in statements such as *The dog was on its run, Jim participated in the 10K fun run,* and *June got a run in her stocking.* Consider the situation where a single term such as *run* can become more than one possible tagged term when tagged, and each of the terms is to be used positively or negatively by the retrieval or filtering system.

As an example, consider a textile manufacturer who is searching for studies on how to stop runs in synthetic fabrics. One system has untagged terms, and *run* occurs with probability .05 in all documents and .2 in relevant documents. We can thus compute $\mathcal{A} = (1 - .2 + .05)/2 = .425$. A second database, containing exactly the same documents and has terms tagged as to whether they are nouns or verbs. The term complex *run-noun* occurs with probability .05 in all documents and probability .5 in relevant documents, thus $\mathcal{A} = .37$. The term complex *run-verb* occurs with probability .05 in the set of all documents and probability of .01 in relevant documents, thus $\mathcal{A} = .57$.

In this case, one $\mathcal{A}$ for a tagged term is better (lower) than the $\mathcal{A}$ for the untagged term, while the other $\mathcal{A}$ for a tagged term is worse than the $\mathcal{A}$ is for the untagged term. An additional method useful for studying $\mathcal{A}$, when there are a number of queries, is to examine the expected A:

$$E(\mathcal{A}) = \sum \Pr(i)\mathcal{A}_i$$

where i is taken here over the set of possible parts-of-speech. We limited ourselves to one untagged term. In the preceding example, if *run-verb* occurs with probability .7 and *run-noun* occurs with probability .3, then $E(\mathcal{A}) = .3(.37) + .7(.57) = .51$

Definition 11.1 (Effective tagging) *If the expected $\mathcal{A}$ for a set of taggings is less that the $\mathcal{A}$ for the untagged terms, then the tagging is said to be* effective.

If $E(\mathcal{A})_{tagged} < \mathcal{A}_{untagged}$ (in our example, $.51 < .425$) then tagging is superior, on the average, to having untagged documents and queries. If the inequality doesn't hold (as it doesn't hold in our case), then tagging will provide inferior results, on the average.

Entries like part-of-speech tags may be applied to other types of tagging or additions to terms and documents. For example, the addition of metadata tags can allow type-of-document information to be used in retrieval. One can also use controlled vocabulary terms to improve retrieval performance.

EXERCISE 11.4. [*Research Problem*] Consider several term types, each of which may have several different part-of-speech tags. How does performance vary with the presence or absence of these different tags?

EXERCISE 11.5. [*Research Problem*] Using the results of the previous exercise, how do phrase tags as well as individual part-of-speech tags affect performance. Consider for example a tag *noun phrase*, for a sequence such as *determiner* followed by *noun* for the phrase *the book*. Thus, what is the performance with individual term tags such as *determiner* and *noun* versus performance with these tags and the phrase tags such as *noun phrase*?

11.4 CONTROLLED VOCABULARIES

Documents and multimedia may be represented using any of the naturally occurring features in each document, or the representation may be features selected from a set of *controlled* terms. A controlled vocabulary is a set of terms or phrases that contains unique ways of representing concepts occurring in the database, avoiding the "crisscross of many-one and one-many relationships between words and their referents" (Svenonius, 1986). Some have viewed the user of controlled vocabularies as enhancing the precision and recall of a search (Svenonius, 1986) while others have suggested that in some environments a controlled vocabulary must be viewed as a "precision device and not that of a recall device" (Boyce & McLain, 1989).

Which vocabulary type performs best has long been the subject of experimental work, but it can be addressed analytically as well as experimentally. Deciding when to use one vocabulary type or another may thus be studied by noting parameter values for different databases and the nature of searches, noting whether or not the conditions of the analytically determined rules are met.

For our analytic work below, we will usually address individual terms or phrases rather than groups of terms or multiple phrases. Thus, when studying a query such as *dogs and rabies* the relationships and dependencies between these two concepts will be ignored. Put more positively, the terms will be treated as though they are statistically independent. The separate concepts will be treated separately, and the performance will be studied for queries as though there was a single term or phrase in each query, thus allowing us to isolate problems. One can quickly lose sight of underlying mechanisms, and our focus here is explicitly on understanding when controlled vocabularies provide better representations for retrieval than are obtained with uncontrolled vocabularies.

Controlled terms can be used to represent equivalent or similar meanings, different spellings or abbreviations or translations, as well as hierarchical relationships. These

concepts have multiple ways they can be represented by a single text string in a controlled vocabulary. The binary term frequency of a controlled vocabulary term d_c in a document may be computed as the maximum of a set of n other uncontrolled synonyms in a document, thus $d_c = max(d_1, d_2, d_3, \ldots, d_n)$. This assumes that any occurrence of $d_i,\ i = 1, 2, \ldots, n$ represents an occurrence of the idea underlying d_c. The controlled vocabulary expression *house pets* might be assigned the maximum binary frequency in a document from the uncontrolled terms *dogs*, *cats*, *tropical fish*, and *parakeets*.

Homonyms (and more properly homographs) are words with different meanings that sound (or are written) the same. The distinction between the two is important, as *two, too,* and *to* are homonyms and sound the same but are written differently. Homographs may best be understood as a written text string with more than one definition or meaning. When a searcher wishes to find a particular meaning or concept, it may be necessary to retrieve all documents containing the meaning, as represented by a particular term, and all the documents containing homographs for that term. Searching using a controlled vocabulary would help the user exclude non-relevant documents containing homographs.

Performance with Controlled Vocabularies

We can easily describe the performance of retrieval with a single term or concept. Given a choice of using one controlled vocabulary term (or a phrase, which may be treated as a single unit) or a single free-text, uncontrolled term, which should be used? Humans use their own rules for searching, with professionals using more sophisticated methods that take into account many variables and nuances not addressed by an analytic model, resulting in superior retrieval performance (Fidel, 1992). The following rule can be used by searchers to improve performance:

> Choose to use a controlled vocabulary term if and only if the performance using the controlled term is better than that obtained with the free-text, uncontrolled term.

Performance with a controlled vocabulary query is superior to performance with an uncontrolled vocabulary query when

$$\mathcal{A}_c \leq \mathcal{A}_u, \tag{11.8}$$

where A_c denotes the A value for controlled vocabulary and A_u denotes the A value for uncontrolled vocabulary.

Theorem 11.6 (Controlled vocabulary & performance) *Controlled vocabulary improves performance when*

$$t_c - p_c \leq t_u - p_u,$$

where the subscripts u and c represent uncontrolled and controlled vocabulary parameters, respectively.

A transformation may illuminate the problem further:

Corollary 11.4 *A controlled vocabulary will outperform an uncontrolled vocabulary when*

$$t_c - t_u \leq p_c - p_u$$

that is, performance will be better using a controlled vocabulary when the difference between the rate of occurrence of the term between documents with controlled vocabularies and terms without controlled vocabularies is less than the difference between the rates of occurrences (in relevant documents only) for the two vocabulary types.

When considering the entire controlled vs. uncontrolled vocabulary, instead of just a single term, one may wish to use a controlled vocabulary only when, on an average, it decreases $\mathcal{A}$ when compared to what is obtained with uncontrolled vocabularies. This may be expressed as

$$E(\mathcal{A}_c) \leq E(\mathcal{A}_u). \tag{11.9}$$

EXERCISE 11.6. [*Moderate*] Arbitrarily select one term related to hobbies or forms of entertainment you enjoy. Find the controlled vocabulary counterpart for this term by looking up the term in the list used to provide controlled vocabulary for a library or indexing service, such as the *Library of Congress Subject Headings*. If the controlled and uncontrolled terms are the same, start over. Search an online catalog for all occurrences of documents with either the controlled or the uncontrolled terms and judge their relevance. Using Theorem 11.6 and parameter values for this dataset, will the controlled vocabulary perform better than the uncontrolled vocabulary?

EXERCISE 11.7. [*Research Problem*] Assume that there is a single underlying meaning for all occurrences of any term within a set of terms that are synonymous with each other. Treat this underlying meaning as a tag of sorts. Provide conditions under which the use of synonymous terms increases or decreases performance. Under what condition would using synonymous terms results in performance inferior or superior to that provided by a single controlled vocabulary term?

11.5 MULTIPLE TERMS AND GRAMMATICAL STRUCTURES

Instead of using statistically independent individual terms when computing document probabilities, term pairs, triples, etc. may be used. The occurrence of n consecutive linguistic units, such as terms, syllables, or letters, may be treated as a unit referred to as an n-gram. Computing the probabilities for these units may be valuable in that structural information is statistically captured. Implicit in limiting the system's focus to windows of a particular size is that one term is more likely to be related to a neighboring term than it is to be related to a distant term (Haas & Losee, 1994; Jacquemin, 1996).

Neighboring terms may be treated so that the order of the terms in the data is considered or so the order that naturally occurs is ignored when computing probabilities. The latter is usually the case in statistical information retrieval models. Treating terms without regard to their order will allow for more accurate estimates to be made, but this is particularly difficult with larger n-grams (Losee, 1994a). Several methods are available for incorporating dependencies or term ordering, including the Bahadur Lazarsfeld Expansion, Bayesian networks, Hidden Markov Models, and autocorrelations, as discussed in Chapter 7.

The probabilistic relationships between terms or phrases may be estimated from the parts-of-speech of the terms in question, as well as from the parts-of-speech of neighboring terms. Different pairs of grammatical components will have different correlations and knowledge of these may be used as estimators of the degree of asso-

ciation between two specific words. If, for example, there is a probability of .25 that an adjective is followed immediately by a noun, this might be used as an estimate of the relationships between *ecru* and *spaceship,* a term pair that may occur with such infrequency that the accurate estimation of the relationship between the two terms is almost impossible.

Non-topical characteristics such as writing *style* may be learned statistically, given the availability of data containing this information. Learning the statistical relationships between grammatical components, as represented by part-of-speech tags, can allow a system to capture knowledge about structural relationships. Consider prose such as:

> See Jane run.
> Run Jane, run.
> Watch Spot go.
> Go Spot, go.

This is easily identified by most adults as rather boring prose written for beginning readers. The grammatical structure here is highly regular, and the reader could probably develop a simple parody of this kind of prose for almost any subject matter in a few moments. These grammatical patterns are easily represented statistically, and when one wishes to retrieve documents of this grammatical type, one can discriminate using probabilities describing the structures in relevant documents and in all documents. Similarly, one can discriminate based on a wide variety of non-topical aspects of documents such as linguistic characteristics and semantic styles. The more subtle are the aspects, the more data is required from which to learn with a given degree of discriminatory accuracy.

Some grammatical constructs occur with a higher rate when using discipline-specific vocabularies than when using more general vocabularies (Losee, 1996). If we believe that these constructs represent those portions of a document that are most likely to be found relevant by the user, then structures with these correlations may provide estimators for correlations between subject-bearing terms in relevant documents. The expected mutual information measure (EMIM) may prove computationally useful when describing these inter-term relationships. Remember that it is otherwise difficult to accurately estimate probabilities in relevant documents for complex structures.

EXERCISE 11.8. [*Difficult*] The adjective *ecru* occurs in 2 of 4 relevant documents that have been retrieved and the noun *spaceship* occurs in 3 of the 4. If the correlation between an adjective and a following noun is 1/4, what is a reasonable estimate of the probability of the phrase *ecru spaceship* occurring in a relevant document?

Frames, Slots, and Cases

To study groups of terms in useful relationships, viewing the term clusters as *frames* (Metzing, 1979) or *cases* (Fillmore, 1977) or *case frames* (Charniak & McDermott, 1985) may prove helpful. These structures provide a means of representing the structure or meaning inherent in natural language in a standardized and structured way. A case or a frame represents a scenario, a set of events, actions, or places that can be described. A frame contains *slots,* representing characteristics of the scenario. Slots may have values derived from a natural language statement or may have default values. For example, a frame for the Christmas scenario might have a default estab-

lished that if there is a *Christmas tree,* then this tree is assumed to be *coniferous*, *decorated*, and *under twenty-feet tall.*

Assume that a query may be treated as a single frame or case. Retrieval may be based upon either an exact match between the query and a case representing a document, with the matching being between entire frames or specified parts of frames. The document cases may be ranked based upon probabilistic considerations, with the query case providing initial knowledge about the parameters of query features.

Frames also may be ranked for presentation to the searcher in order of their retrieval status value, associated with the degree of relationship between the query and the document frames. Different slots within the frame may have different discrimination values, depending upon the relationships between slot value frequencies in the slots and the query.

The probability of a particular frame with a given set of slot values may be computed from historical data. This probability may be decomposed into two probabilities, the probability that the type of frame found is the type that did occur, and the probability the frame will have the values that it has, given that it is this type of frame that has been found. The product of these two probabilities is the unconditional probability that a particular frame occurs. The probabilities associated with frames and cases may be used to compute the expected performance. More traditional grammars may also be used in computing expected performance, as is discussed below.

11.6 THE STRUCTURE OF STATEMENTS

Much of modern linguistic theory is based on grammars taken from the Chomsky hierarchy of types of formal languages. Grammars represent the structure of sentences and the relationships between terms and are thus important components in the production of relationships that affect retrieval performance. Chomsky, who argued in 1969 that "It must be recognized that the notion of a 'probability of a sentence' is an entirely useless one, under any interpretation of this term" (Ney, 1997), moved the assignment of part-of-speech tags away from the statistical, although tagging has been moving back toward the statistical as a result of recent successes. For Chomsky, a grammar is a 4-tuple $G =< T, z, \mathcal{R}, S >$ where T is the set of non-terminal symbols in the language, e.g. $A, B, \ldots, Y$, the set z is the set of terminal symbols, e.g., $a, b, \ldots, y$, m grammatical rewrite rules $\mathcal{R} = r_1, r_2, \ldots, r_m$, and S is the starting symbol.

A rule might be written as $A \rightarrow BCD$, indicating that non-terminal symbol A is replaced by ("rewritten as") the non-terminal symbols B, C, and D, in that order. Non-terminal symbols such as A and B are often rewritten eventually as terminal symbols, tokens occurring in a sentence.

The Chomsky hierarchy is a ranking of grammar types, with each type being precisely described by a set of constraints on grammars of that type. A grammar of type n produces or describes a language of type n. A type n language is also a language of type $n-1$. As one moves from a type n grammar to a type $n-1$ grammar, the restrictions on the grammar are relaxed, with a type 0 grammar having the fewest restrictions.

Regular (Type 3) Grammars

The most restrictive set of rules are assigned to a *regular finite state*, or *type 3* grammar. The right hand side (RHS) of each rule contains one terminal symbol and optionally a non-terminal symbol. Thus, a type 3 grammar allows the replacement string in a production to be restricted to either terminal symbols or a terminal symbol immediately followed by a single nonterminal symbol. Sample valid rules would then include

$$X \to y$$
$$X \to yZ.$$

However, a rule such as

$$A \to Ab$$

is not allowed.

Context Free (Type 2) Grammars

Unlike the type 3 grammar, which limits the right hand side of a rule to a terminal symbol, optionally followed by a non-terminal symbol, a *type 2* or *context free* grammar allows any combination of non-terminal and terminal symbols on the right hand side of a production rule. The right hand side may also be null. The left hand side of a type 2 rule is always a single, non-terminal symbol. Thus, in a rule of the form $\alpha C\beta \to \alpha d\beta$, both α and β must be null.

Context free production rules might look like:

$$A \to Ab$$
$$A \to AB$$
$$A \to \phi$$

Context Sensitive (Type 1) Grammars

A *context sensitive*, *context dependent*, or *Type 1* grammar allows for rules such as $\alpha A\beta \to \alpha\psi\beta$ where α and β are possibly null sequences. Because there can be a "context" allowed on the left hand side of a rule, it is referred to as context-sensitive grammar. The variable ψ represents a non-empty sequence of symbols, removing the capability of a production rule to delete a symbol. The length of the left hand side of this rewrite rule is always less than or equal to the length of the RHS.

Context free languages that don't have the null string are a proper subset of the context sensitive languages. The following thus are valid rules in a context sensitive grammar

$$RstU \to Rstu$$
$$R \to Rs$$

while the following is not valid

$$Rt \rightarrow t.$$

Phrase Structure (Type 0) Grammars

The most general type of grammar is a phrase structure grammar, which allows the context sensitive capability of a type 1 grammar and allows the null-string to be produced in the language.

Grammar Types and Natural Language

While any of these types of grammars could be the basis for most natural languages, context free grammars appear to be satisfactory approximations of a human's grammar in most situations. Partee, Meulen and Wall (Partee, Meulen, & Wall, 1990) note that "it took nearly thirty years to find one completely convincing example of a natural language which was not context free." Adding some of the features of a context sensitive grammar are necessary in some cases, and a number of mildly context sensitive grammars have been proposed, including index grammars and head grammars, that may prove useful when limiting the application of (otherwise context-free) rules to specific terms or situations.

Because context-free grammars are adequate in many situations and are mathematically far more tractable than are other grammars, we will examine the probabilities of rules assuming that the grammar is consistent with the assumptions of a context free grammar.

The Probability of a Parse

It is easy to compute the probability of a fully tagged statement, produced by a parser consistent with a context free grammar (Charniak, 1993; Fu, 1982; Lee & Fu, 1972). A fully tagged and parsed sentence is the original set of terms along with the full parse tree. This is the set of all tags (non-terminal symbols) used in the parse and terms (terminal symbols) from the statement. Computing the probability of a parse given that it is in the set of relevant documents or the probability of a parse from the set of all documents allows us to use the analytic methods to compute the expected performance for a retrieval system with documents written in a natural language.

Each production rule has an assigned probability, computed or estimated (Fu, 1982) so that the sum of probabilities associated with the set of rules having a given left hand side x, such as

$$\begin{aligned} x &\rightarrow a \\ x &\rightarrow b \\ x &\rightarrow \ldots \\ x &\rightarrow n, \end{aligned}$$

is 1. The probability of a given parse is the product of the probability of each of the rules that is applied in producing the full parse. If the parsings of different statements are statistically independent, then the probability of the entire parse

Table 11.2. Sample sentences, part-of-speech tags and relevance values for 10 statements. Probabilities are for applied rewrite rules for the sentences. LHS represents the left hand side of a rewrite rule, and RHS represents the right hand side of a rule.

Sentence	*POS Tags*	*Rel.*	*Sentence*	*POS Tags*	*Rel.*
v w	A B	R	x w	A B	R
y w	A B	R	v z	A B	N
v y	A C	N	w v	B A	N
w z	B C	N	v w	A C	N
y w	A B	N	w x	C B	N

Rule	$\Pr(RHS\|LHS)$	$\Pr(RHS\|LHS, Rel)$
$S \rightarrow RL$	1	1
$L \rightarrow A$	7/10	3/3
$L \rightarrow B$	2/10	0
$L \rightarrow C$	1/10	0
$R \rightarrow A$	1/10	0
$R \rightarrow B$	6/10	3/3
$R \rightarrow C$	3/10	0
$A \rightarrow v$	5/8	1/3
$A \rightarrow x$	1/8	1/3
$A \rightarrow y$	2/8	1/3
$B \rightarrow w$	6/8	3/3
$B \rightarrow x$	1/8	0
$B \rightarrow z$	1/8	0
$C \rightarrow w$	2/4	—
$C \rightarrow y$	1/4	—
$C \rightarrow z$	1/4	—

may be computed by multiplying the probabilities of the independent parses. Text filtering and retrieval performance may be predicted using techniques incorporating term dependencies described in Chapter 8. Computing the performance with these methods, we may make a number of performance claims about systems incorporating natural language capabilities.

EXERCISE 11.9. [*Moderate*] Develop a set of syntactic rules for the following arithmetic statements: $3 + 2 = 5$, $12 - 4 = 8$, $3 + 3 + 4 = 10$, and $3 = 3$. What is the probability for each rule given this data set?

11.7 PERFORMANCE WITH MULTIPLE SYNTACTIC TAGS: A CASE STUDY

Term Independence Performance

Consider a query that consists of only the term w. Using the data in Table 11.2, we find that among the ten sentences, there are 8 sentences with the query term w (so

$t = .8$) and from among the 3 relevant sentences all have w (thus $p = 1$). Thus, using $\mathcal{A} = (1 - p + t)/2$, $\mathcal{A} = (1 - 1 + .8)/2 = .4$.

Using Syntactic Structures

Consider a query for w tagged so that it as at the end of the sentence, in the rightmost position, e.g. $R - w$, with optionally any preceding non-terminal symbol. Here R represents "right." Looking for w in sentence final position, we find that this occurs five times in the ten sentences, and thus $t = .5$, while among the relevant sentences, it occurs in all the relevant sentences. We thus find that $\mathcal{A} = (1 - 1 + .5)/2 = .25$. Using linguistic tagging in this case provides a marked improvement over the results obtained with the binary independent model, from $\mathcal{A} = .4$ improving to $\mathcal{A} = .25$.

Now, let us assume that the query is further refined so that it is $R - B - w$, that is, w of part-of-speech B of part-of-speech R. The probability that we achieve this from among the set of all sentences is $t = \Pr(R - B - w) = 4/10$. Using similar techniques, $\Pr(R - B - w|rel) = 1$. The optimal performance may thus be computed as $\mathcal{A} = (1 - 1 + 4/10)/2 = .2$, an improvement over the previous case.

Clearly, the more grammatical knowledge one has of a term in this example, the better will be the filtering performance.

11.8 EVALUATING GRAMMAR QUALITY WITH RETRIEVAL PERFORMANCE

Evaluating one or more grammars that might be used with a filtering system can allow one to select a good, or possibly the best grammar, for a particular application or family of applications. We can describe the performance relationship between grammars by noting the ASL values obtained with two grammars for either one query, or the ASL for two grammars taken over a range of queries. Some grammars are expected to be better than others for all cases where all query terms are positive or neutral discriminators. For example, when grammar $\mathcal{G}_i$ can discriminate between terms that a second grammar $\mathcal{G}_j$ can't, and the grammars are otherwise equivalent, then $\mathcal{G}_i$ is expected to be superior to $\mathcal{G}_j$.

Definition 11.2 (Grammar Superiority) *Given the ASL of grammar $\mathcal{G}$, denoted as $ASL(\mathcal{G})$, we denote the superiority or equality of grammar i over grammar j as*

$$\mathcal{G}_{i,x} \overset{*}{\geq} \mathcal{G}_{j,x}$$

for domain x when $ASL(\mathcal{G}_{i,x}) \leq ASL(\mathcal{G}_{j,x})$.

Theorem 11.7 (Transitivity of grammar superiority) *If $\mathcal{G}_{i,x} \overset{*}{\geq} \mathcal{G}_{j,x}$ and $\mathcal{G}_{j,x} \overset{*}{\geq} \mathcal{G}_{k,x}$ then $\mathcal{G}_{i,x} \overset{*}{\geq} \mathcal{G}_{k,x}$ for all i, j, k, and x.*

Informally, we may describe one grammar as *superior* to another grammar for a given query if the $\mathcal{A}$ values for the performatively superior grammar are less than the $\mathcal{A}$ values for the inferior grammar. A *performance hierarchy* of grammars is as follows:

1. **Superiority occurs in a domain**
   $\mathcal{G}_i$ is *performatively superior* to (or equal to) $\mathcal{G}_j$ in domain x when $\mathcal{G}_{i,x} \overset{*}{\geq} \mathcal{G}_{j,x}$ with the expected ASL for the grammars taken over domain x.

2. **Superiority occurs in all cases studied**
   $\mathcal{G}_i$ is *universally superior* to (or equal to) $\mathcal{G}_j$ when $\mathcal{G}_i$ is *performatively superior* to (or equal to) $\mathcal{G}_j$ in domain x, and x is the universe of all extant domains. This is an empirical superiority.

3. **Superiority is necessary**
   $\mathcal{G}_i$ is *necessarily superior* to $\mathcal{G}_j$ when it is necessary that $\mathcal{G}_{i,x} \overset{*}{\geq} \mathcal{G}_{j,x}$ for all possible domains x.

Consider the case where there are a set of queries, some with *run* as a noun and some with *run* as a verb. Assume that we have a grammar $\mathcal{G}_0$ which has no part-of-speech tagging, essentially a null grammar. We also have $\mathcal{G}_{nv}$ which can perfectly parse all sentences and tag *n*ouns and *v*erbs as such. In some situations, queries will specify that in the case of ambiguous terms, the verb form is the one that is desired or the nominal form is the one that is sought. This would lead to improved performance for many of the queries that can use the partial-disambiguation that is available through use of parts-of-speech. One thus expects that $\mathcal{G}_{nv} \overset{*}{\geq} \mathcal{G}_0$ and we can say that $\mathcal{G}_{nv}$ is necessarily superior to $\mathcal{G}_0$ if query terms are positive discriminators.

Further, if we have a grammar $\mathcal{G}_{n_{123}v}$ that can distinguish between verbs and nouns and can also distinguish three different kinds of nouns, and these noun types occur in some queries, we would expect $\mathcal{G}_{n_{123}v}$ to be necessarily superior to $\mathcal{G}_{nv}$ when pairs of query terms and part-of-speech tags are positive discriminators.

A weaker case, where only performative superiority would hold, would be a situation where a grammatical tagger that only tagged nouns (leaving all other terms as being the same) was shown to produce better retrieval on an average than a tagger that only tagged adjectives (leaving all other terms as being the same). The performative difference between these two is an empirical question (at present) and we than thus say that the noun tagging grammar $\mathcal{G}_n$ is only performatively superior to the adjective tagging grammar $\mathcal{G}a$.

Note that necessary superiority, or conditions under which necessary superiority holds, may be determined using analytic methods.

Language Supporting Actions

Our ability to analytically study the retrieval and filtering of natural language statements can lead us to a more general model of how natural languages allow us to discriminate between linguistic expressions (messages) that facilitate human communication and decision making. Consider a message M of benefit b to a human. We assume that messages having benefit b or over are considered relevant and messages with benefit below b to be non-relevant. We assume that a particular benefit is due to an association between the message and a physical action that may be taken with associated benefit b. This message has the average position of $\mathcal{A}$.

Assume that we have a set of part-of-speech tags T for the language and a set of syntactic rules $\mathcal{R}$. The grammar $\mathcal{G}$ is a function described earlier in the chapter. Let us also assume that there exists an error function $\mathcal{H}$ that is due to the human

constraints on the amount and rate at which humans can learn and currently operate. The $\mathcal{H}$ function is a source of error, as is a $\mathcal{Q}$ less than 1. This $\mathcal{H}$ function is assumed to be non-linearly related to the number and complexity of syntactic rules $\mathcal{R}$ and tags T. The set of tags T and the assignments of tags to message terms is such that a certain level of filtering performance, $\mathcal{A}$, is obtained.

Conjecture 11.1 (Optimal language) *An optimal language, given a world where benefits b are associated with receiving messages M, is such that parameters $\mathcal{R}$ and T minimize*

$$E\Big(ASL\Big(A\big(\mathcal{G}(T,\mathcal{R})\big),\mathcal{H}(T,\mathcal{R})\Big),M\Big) \tag{11.10}$$

over the set of messages encountered.

Consider what happens when the number of part-of-speech (POS) tags and syntactic rules increases. As the number of part-of-speech tags increase, $\mathcal{A}$ will decrease (improve) as the ability to discriminate between potentially ambiguous terms is increased. However, performance will have a component that tends toward decreasing performance as the error function returns a greater error value that is proportional to the increased number of part-of-speech tags, system complexity, and the increased chance for human error due to the difficulty associated with learning a more complex language.

A human system that must discriminate and make decisions based on natural language will discriminate best when using an optimal language. We may assume that an optimal language is achieved when the ASL value in Equation 11.10 is minimized. This is obtained at a given syntactic complexity and a given number of part-of-speech tags and represents the break-even point between these two conflicting factors. At this point, the performance decrease in error rate due to increased language complexity is the same as the performance increase due to the increased value of discrimination capability that is available with increasingly complex linguistic structures.

The error function for humans may be learned if natural language is assumed to be optimized for different applications and sublanguages, e.g. children vs. adults, folksy chatting vs. technical reports, etc. (Sager, 1981a). By learning the functional values for several different language types, the error function may be estimated and its general characteristics understood.

11.9 DISCUSSION

The difficulties associated with satisfactorily incorporating natural language processing into retrieval systems present unique challenges to those estimating filtering and retrieval performance. Part-of-speech tags can help filtering systems separate relevant documents from non-relevant documents. A rule was proposed (Equation 11.1) using these tags that suggested that improvement would occur for retrieving individual terms for certain ranges of parameter values associated with the tagging process. Tagging with multiple terms and the stemming of suffixes similarly may improve performance.

The relationships between terms may improve filtering system performance. Grammars may be used to parse statements, providing part-of-speech tags and implicitly providing information about the relationships between terms. Methods described

in Chapter 7 can use these term relationships to provide improved document ranking. The case study provided in this chapter shows how increasing the amount of linguistically provided information can improve performance.