8 *Lexical Acquisition*

THE TOPIC of chapter 5 was the acquisition of *collocations*, phrases and other combinations of words that have a specialized meaning or some other special behavior important in NLP. In this chapter, we will cast our net more widely and look at the acquisition of more complex syntactic and semantic properties of words. The general goal of lexical acquisition is to develop algorithms and statistical techniques for filling the holes in existing machine-readable dictionaries by looking at the occurrence patterns of words in large text corpora. There are many lexical acquisition problems besides collocations: selectional preferences (for example, the verb *eat* usually takes food items as direct objects), subcategorization frames (for example, the recipient of *contribute* is expressed as a prepositional phrase with *to*), and semantic categorization (what is the semantic category of a new word that is not covered in our dictionary?). While we discuss simply the ability of computers to learn lexical information from online texts, rather than in any way attempting to model human language acquisition, to the extent that such methods are successful, they tend to undermine the classical Chomskyan arguments for an innate language faculty based on the perceived poverty of the stimulus.

LEXICAL ACQUISITION

Most properties of words that are of interest in NLP are not fully covered in machine-readable dictionaries. This is because of the productivity of natural language. We constantly invent new words and new uses of old words. Even if we could compile a dictionary that completely covered the language of today, it would inevitably become incomplete in a matter of months. This is the reason why lexical acquisition is so important in Statistical NLP.

LEXICAL
LEXICON

A brief discussion of what we mean by *lexical* and the *lexicon* is in order. Trask (1993: 159) defines the lexicon as:

LEXICAL ENTRIES

> That part of the grammar of a language which includes the *lexical entries* for all the words and/or morphemes in the language and which may also include various other information, depending on the particular theory of grammar.

The first part of the definition ("the lexical entries for all the words") suggests that we can think of the lexicon as a kind of expanded dictionary that is formatted so that a computer can read it (that is, machine-readable). The trouble is that traditional dictionaries are written for the needs of human users, not for the needs of computers. In particular, quantitative information is completely missing from traditional dictionaries since it is not very helpful for the human reader. So one important task of lexical acquisition for Statistical NLP is to augment traditional dictionaries with quantitative information.

The second part of the definition ("various other information, depending on the particular theory of grammar") draws attention to the fact that there is no sharp boundary between what is lexical information and what is non-lexical information. A general syntactic rule like S → NP VP is definitely non-lexical, but what about ambiguity in the attachment of prepositional phrases? In a sense, it is a syntactic problem, but it can be resolved by looking at the lexical properties of the verb and the noun that compete for the prepositional phrase as the following example shows:

(8.1) a. The children ate the cake with their hands.

b. The children ate the cake with blue icing.

We can learn from a corpus that eating is something you can do with your hands and that cakes are objects that have icing as a part. After acquiring these lexical dependencies between *ate* and *hands* and *cake* and *icing*, we can correctly resolve the attachment ambiguities in example (8.1) such that *with their hands* attaches to *ate* and *with blue icing* attaches to *cake*.

In a sense, almost all of Statistical NLP involves estimating parameters tied to word properties, so a lot of statistical NLP work has an element of lexical acquisition to it. In fact, there are linguistic theories claiming that all linguistic knowledge is knowledge about words (Dependency Grammar (Mel′čuk 1988), Categorial Grammar (Wood 1993), Tree Adjoining Grammar (Schabes et al. 1988; Joshi 1993), 'Radical Lexicalism' (Karttunen 1986)) and all there is to know about a language is the lexicon, thus completely dispensing with grammar as an independent entity. In general, those properties that are most easily conceptualized on the level of the individual word are covered under the rubric 'lexical acquisition.' We have devoted separate chapters to the acquisition of collocations and word sense disambiguation simply because these are self-contained and warrant separate treatment as central problems in Statistical NLP. But they are as much examples of lexical acquisition as the problems covered in this chapter.

The four main areas covered in this chapter are verb subcategorization (the syntactic means by which verbs express their arguments), attachment ambiguity (as in example (8.1)), selectional preferences (the semantic characterization of a verb's arguments such as the fact that things that get eaten are usually food items), and semantic similarity between words. However, we first begin by introducing some evaluation measures which are commonly used to evaluate lexical acquisition methods and various other Statistical NLP systems, and conclude with a more in-depth discussion of the significance of lexical acquisition in Statistical NLP and some further readings.

8.1 Evaluation Measures

An important recent development in NLP has been the use of much more rigorous standards for the evaluation of NLP systems. It is generally agreed that the ultimate demonstration of success is showing improved performance at an application task, be that spelling correction, summarizing job advertisements, or whatever. Nevertheless, while developing systems, it is often convenient to assess components of the system on some artificial performance score (such as perplexity), improvements in which one can expect to be reflected in better performance for the whole system on an application task.

Evaluation in Information Retrieval (IR) makes frequent use of the notions of precision and recall, and their use has crossed over into work on evaluating Statistical NLP models, such as a number of the systems discussed in this chapter. For many problems, we have a set of targets (for example, targeted relevant documents, or sentences in which a word has a certain sense) contained within a larger collection. Our system then decides on a selected set (documents that it thinks are relevant, or sentences that it thinks contain a certain sense of a word, etc.). This situation is shown in figure 8.1. The selected and target groupings can be thought

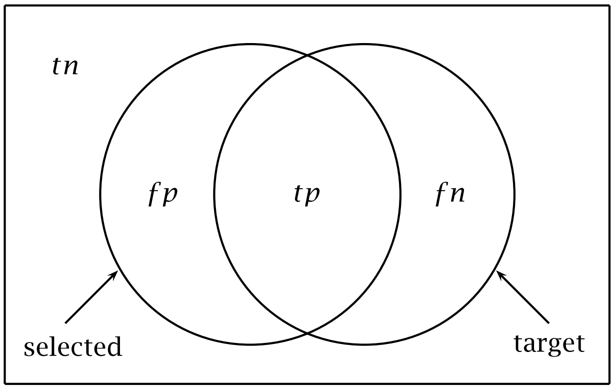

Figure 8.1 A diagram motivating the measures of precision and recall. The areas counted by the figures for true and false positives and true and false negatives are shown in terms of the target set and the selected set. Precision is $tp/|\text{selected}|$, the proportion of target (or correct) items in the selected (or retrieved) set. Recall is $tp/|\text{target}|$, the proportion of target items that were selected. In turn, $|\text{selected}| = tp + fp$, and $|\text{target}| = tp + fn$).

of as indicator random variables, and the joint distribution of the two variables can be expressed as a 2×2 contingency matrix:

(8.2)

	Actual	
System	target	¬ target
selected	tp	fp
¬selected	fn	tn

TRUE POSITIVES
TRUE NEGATIVES
FALSE POSITIVES
TYPE II ERRORS
FALSE NEGATIVES
TYPE I ERRORS
PRECISION

The numbers in each box show the frequency or count of the number of items in each region of the space. The cases accounted for by tp (*true positives*) and tn (*true negatives*) are the cases our system got right. The wrongly selected cases in fp are called *false positives*, *false acceptances* or *Type II errors*. The cases in fn that failed to be selected are called *false negatives*, *false rejections* or *Type I errors*.

Precision is defined as a measure of the proportion of selected items that the system got right:

$$\text{precision} = \frac{tp}{tp + fp} \tag{8.3}$$

RECALL

Recall is defined as the proportion of the target items that the system selected:

$$\text{recall} = \frac{tp}{tp + fn} \tag{8.4}$$

In applications like IR, one can generally trade off precision and recall (one can select every document in the collection and get 100% recall but very low precision, etc.). This tradeoff can be plotted in a precision-recall curve, as we illustrate in section 15.1.2. Sometimes such a tradeoff doesn't make as much sense in NLP applications, but in any situation where there are some items that one is more sure of than others (such as in subcategorization frame learning in section 8.2), the same opportunities for trading off precision vs. recall exist.

For this reason it can be convenient to combine precision and recall into a single measure of overall performance. One way to do this is the *F measure*, a variant of the *E measure* introduced by van Rijsbergen (1979: 174), where $F = 1 - E$. The F measure is defined as follows:

F MEASURE

E MEASURE

$$F = \frac{1}{\alpha \frac{1}{P} + (1 - \alpha) \frac{1}{R}} \tag{8.5}$$

where P is precision, R is recall and α is a factor which determines the weighting of precision and recall. A value of $\alpha = 0.5$ is often chosen for equal weighting of P and R. With this α value, the F measure simplifies to $2PR/(R + P)$.

A good question to ask is: "Wait a minute, in the table in (8.2), $tp + tn$ is the number of things I got right, and $fp + fn$ is the number of things I got wrong. Why don't we just report the percentage of things right or the percentage of things wrong?" One can do that, and these measures are known as *accuracy* and *error*. But it turns out that these often aren't good measures to use because in most of the kinds of problems we look at tn, the number of non-target, non-selected things, is huge, and dwarfs all the other numbers. In such contexts, use of precision and recall has three advantages:

ACCURACY

ERROR

- Accuracy figures are not very sensitive to the small, but interesting numbers tp, fp, and fn, whereas precision and recall are. One can get extremely high accuracy results by simply selecting nothing.

- Other things being equal, the F measure prefers results with more true positives, whereas accuracy is sensitive only to the number of errors. This bias normally reflects our intuitions: We are interested in finding things, even at the cost of also returning some junk.

	tp	fp	fn	tn	Prec	Rec	F	Acc
(a)	25	0	125	99,850	1.000	0.167	0.286	0.9988
	50	100	100	99,750	0.333	0.333	0.333	0.9980
	75	150	75	99,700	0.333	0.500	0.400	0.9978
	125	225	25	99,625	0.357	0.833	0.500	0.9975
	150	275	0	99,575	0.353	1.000	0.522	0.9973
(b)	50	0	100	99,850	1.000	0.333	0.500	0.9990
	75	25	75	99,825	0.750	0.500	0.600	0.9990
	100	50	50	99,800	0.667	0.667	0.667	0.9990
	150	100	0	99,750	0.600	1.000	0.750	0.9990

Table 8.1 The F measure and accuracy are different objective functions. The table shows precision, recall, F measure (with $\alpha = 0.5$) and accuracy scores for certain selections of some number of items from out of a collection of 100,000 items of which 150 are genuine targets. The upper series (a) shows increasing F measure values, but decreasing accuracy. The lower series (b) shows identical accuracy scores, but again increasing F measure values. The bias of the F measure is towards maximizing the true positives, while accuracy is sensitive only to the number of classification errors.

- Using precision and recall, one can give a different cost to missing target items versus selecting junk.

Table 8.1 provides some examples which illustrate how accuracy and the F measure (with $\alpha = 0.5$) evaluate results differently.

FALLOUT

A less frequently used measure is *fallout*, the proportion of non-targeted items that were mistakenly selected.

$$\text{fallout} = \frac{fp}{fp + tn} \tag{8.6}$$

Fallout is sometimes used as a measure of how hard it is to build a system that produces few false positives. If the number of non-targeted items is very large, then low precision due to large fp may be unavoidable because with a large background population of non-targeted items, it is unavoidable that some will be miscategorized.

ROC CURVE

In some fields of engineering recall-fallout trade-offs are more common than precision-recall trade-offs. One uses a so-called *ROC curve* (for *receiver operating characteristic*) to show how different levels of fallout (false positives as a proportion of all non-targeted events) influence recall

Frame	Functions	Verb	Example
NP NP	subject, object	greet	She greeted me.
NP S	subject, clause	hope	She hopes he will attend.
NP INF	subject, infinitive	hope	She hopes to attend.
NP NP S	subject, object, clause	tell	She told me he will attend.
NP NP INF	subject, object, infinitive	tell	She told him to attend.
NP NP NP	subject, (direct) object, indirect object	give	She gave him the book.

Table 8.2 Some subcategorization frames with example verbs and sentences. (adapted from (Brent 1993: 247)).

or sensitivity (true positives as a proportion of all targeted events). Think of a burglar alarm that has a knob for regulating its sensitivity. The ROC curve will tell you, for a certain rate of false positives, what the expected rate of true positives is. For example, for a false positives rate of being woken up once in a hundred nights with no burglars, one might achieve an expected rate of true positives of 95% (meaning 5% of burglaries will not be detected).

▼ Evaluation measures used in probabilistic parsing are discussed in section 12.1.8, and evaluation in IR is further discussed in section 15.1.2.

8.2 Verb Subcategorization

SUBCATEGORIZE FOR

SUBCATEGORIZATION FRAME

Verbs *subcategorize for* different syntactic categories as we discussed in section 3.2.2. That is, they express their semantic arguments with different syntactic means. A particular set of syntactic categories that a verb can appear with is called a *subcategorization frame*. Examples of subcategorization frames are given in table 8.2. English verbs always subcategorize for a subject, so we sometimes omit subjects from subcategorization frames.

The phenomenon is called subcategorization because we can think of the verbs with a particular set of *semantic* arguments as one category. Each such category has several *subcategories* that express these semantic arguments using different *syntactic* means. For example, the class of verbs with semantic arguments *theme* and *recipient* has a subcategory that expresses these arguments with an object and a prepositional phrase (for example, *donate* in *He donated a large sum of money to the church*), and another subcategory that in addition permits a double-object construction (for example, *give* in *He gave the church a large sum of money*).

Knowing the possible subcategorization frames for verbs is important for parsing. The contrast in (8.7) shows why.

(8.7) a. She told the man where Peter grew up.

b. She found the place where Peter grew up.

If we know that *tell* has the subcategorization frame NP NP S (subject, object, clause), and that *find* lacks that frame, but has the subcategorization frame NP NP (subject, object), we can correctly attach the *where*-clause to *told* in the first sentence (as shown in (8.8a)) and to *place* in the second sentence (as shown in (8.8b)).

(8.8) a. She told [the man] [where Peter grew up].

b. She found [the place [where Peter grew up]].

Unfortunately, most dictionaries do not contain information on subcategorization frames. Even if we have access to one of the few dictionaries that do (e.g., Hornby 1974), the information on most verbs is incomplete. According to one account, up to 50% of parse failures can be due to missing subcategorization frames.[1] The most comprehensive source of subcategorization information for English is probably (Levin 1993). But even this excellent compilation does not cover all subcategorization frames and it does not have quantitative information such as the relative frequency of different subcategorization frames for a verb. And the need to cope with the productivity of language would make some form of acquisition from corpora necessary even if there were better sources available.

A simple and effective algorithm for learning some subcategorization frames was proposed by Brent (1993), implemented in a system called *Lerner*. Suppose we want to decide based on corpus evidence whether verb v takes frame f. Lerner makes this decision in two steps.

- **Cues.** Define a regular pattern of words and syntactic categories which indicates the presence of the frame with high certainty. Certainty is formalized as probability of error. For a particular cue c^j we define a probability of error ϵ_j that indicates how likely we are to make a mistake if we assign frame f to verb v based on cue c^j.

[1] John Carroll, “Automatic acquisition of subcategorization frames and selectional preferences from corpora,” talk given at the workshop “Practical Acquisition of Large-Scale Lexical Information” at CSLI, Stanford, on April 23, 1998.

- **Hypothesis testing.** The basic idea here is that we initially assume that the frame is *not* appropriate for the verb. This is our null hypothesis H_0. We reject this hypothesis if the cue c^j indicate with high probability that our H_0 is wrong.

Cues. Here is the regular pattern that Brent (1993: 247) uses as the cue for the subcategorization frame "NP NP" (transitive verbs):

(8.9) Cue for frame "NP NP":
(OBJ | SUBJ_OBJ | CAP) (PUNC | CC)

where OBJ stands for personal pronouns that are necessarily accusative (or objective) like *me* and *him*, SUBJ_OBJ stands for personal pronouns that can be both subjects and objects like *you* and *it*, CAP is any capitalized word, PUNC is a punctuation mark, and CC is a subordinating conjunction like *if*, *before* or *as*.

This pattern is chosen because it is only likely to occur when a verb indeed takes the frame "NP NP." Suppose we have a sentence like (8.10) which matches the instantiation "CAP PUNC" of pattern (8.9).

(8.10) [...] greet-V Peter-CAP ,-PUNC [...]

One can imagine a sentence like (8.11) where this pattern occurs and the verb does not allow the frame. (The matching pattern in (8.11) is *came*-V *Thursday*-CAP ,-PUNC.) But this case is very unlikely since a verb followed by a capitalized word that in turn is followed by a punctuation mark will almost always be one that takes objects and does not require any other syntactic arguments (except of course for the subject). So the probability of error is very low when we posit the frame 'NP NP' for a verb that occurs with cue (8.9).

(8.11) I came Thursday, before the storm started.

Note that there is a tradeoff between how reliable a cue is and how often it occurs. The pattern "OBJ CC" is probably even less likely to be a misleading cue than "CAP PUNC." But if we narrowed (8.9) down to one reliable instantiation, we might have to sift through hundreds of occurrences of a verb to find the first occurrence with a cue, which would make the test applicable only to the most frequent verbs. This is a problem which we will return to later.

Hypothesis testing. Once the cues for the frames of interest have been defined, we can analyze a corpus, and, for any verb-frame combination, count the number of times that a cue for the frame occurs with the verb. Suppose that verb v^i occurs a total of n times in the corpus and that there are $m \leq n$ occurrences with a cue for frame f^j. Then we can reject the null hypothesis H_0 that v^i does not permit f^j with the following probability of error:

$$p_E = P(v^i(f^j) = 0 | C(v^i, c^j) \geq m) = \sum_{r=m}^{n} \binom{n}{r} {\epsilon_j}^r (1 - \epsilon_j)^{n-r} \tag{8.12}$$

where $v^i(f^j) = 0$ is shorthand for 'Verb v^i does not permit frame f^j,' $C(v^i, c^j)$ is the number of times that v^i occurs with cue c^j, and ϵ_j is the error rate for cue f^j, that is, the probability that we find cue c^j for a particular occurrence of the verb although the frame is not actually used.

Recall the basic idea of hypothesis testing (chapter 5, page 162): p_E is the probability of the observed data if the null hypothesis H_0 is correct. If p_E is small, then we reject H_0 because the fact that an unlikely event occurred indicates assuming H_0 was wrong. Our probability of error in this reasoning is p_E.

In equation (8.12), we assume a binomial distribution (section 2.1.9). Each occurrence of the verb is an independent coin flip for which the cue doesn't work with probability ϵ_j (that is, the cue occurs, but the frame doesn't), and for which it works correctly with probability $1-\epsilon_j$ (either the cue occurs and correctly indicates the frame or the cue doesn't occur and thus doesn't mislead us).[2] It follows that an incorrect rejection of H_0 has probability p_E if we observe m or more cues for the frame. We will reject the null hypothesis if $p_E < \alpha$ for an appropriate level of significance α, for example, $\alpha = 0.02$. For $p_E \geq \alpha$, we will assume that verb v^i does not permit frame f^j.

An experimental evaluation shows that Lerner does well as far as precision is concerned. For most subcategorization frames, close to 100% of the verbs assigned to a particular frame are correctly assigned (Brent 1993: 255). However, Lerner does less well at recall. For the six frames covered by Brent (1993), recall ranges from 47% to 100%, but these numbers would probably be appreciably lower if a random sample of verb types had been selected instead of a random sample of verb tokens,

[2] Lerner has a third component that we have omitted here: a way of determining ϵ_j for each frame. The interested reader should consult (Brent 1993).

a sampling method that results in a small proportion of low-frequency verbs.[3] Since low-frequency verbs are least likely to be comprehensively covered in existing dictionaries, they are arguably more important to get right than high-frequency verbs.

Manning (1993) addresses the problem of low recall by using a tagger and running the cue detection (that is, the regular expression matching for patterns like (8.9)) on the output of the tagger. It may seem worrying that we now have two error-prone systems, the tagger and the cue detector, which are combined, resulting in an even more error-prone system. However, in a framework of hypothesis testing, this is not necessarily problematic. The basic insight is that it doesn't really matter how reliable a cue is as an indicator for a subcategorization frame. Even an unreliable indicator can help us determine the subcategorization frame of a verb reliably if it occurs often enough and we do the appropriate hypothesis testing. For example, if cue c^j with error rate $\epsilon_j = 0.25$ occurs 11 out of 80 times, then we can still reject the null hypothesis that v^i does not permit c^j with $p_E \approx 0.011 < 0.02$ despite the low reliability of c^j.

Allowing low-reliability cues and additional cues based on tagger output increases the number of available cues significantly. As a result, a much larger proportion of verb occurrences have cues for a given frame. But more importantly, there are many subcategorization frames that have no high-reliability cues, for example, subcategorization for a preposition such as *on* in *he relies* ***on*** *relatives* or *with* in *she compared the results* ***with*** *earlier findings*. Since most prepositions occurring after verbs are not subcategorized for, there is simply no reliable cue for verbs subcategorizing for a preposition. Manning's method can learn a larger number of subcategorization frames, even those that have only low-reliability cues.

Table 8.3 shows a sample of Manning's results. We can see that precision is high: there are only three errors. Two of the errors are prepositional phrases (PPs): *to bridge between* and *to retire in*. It is often difficult to decide whether prepositional phrases are arguments (which are subcategorized for) or adjuncts (which aren't). One could argue that *retire* subcategorizes for the PP *in Malibu* in a sentence like *John retires in Malibu* since the verb and the PP-complement enter into a closer relationship than mere adverbial modification. (For example, one can infer that John ended up living in Malibu for a long time.) But the OALD does not list

[3] Each occurrence of a verb in the Brown corpus had an equal chance of appearing in the sample which biases the sample against low-frequency verbs.

Verb	Correct	Incorrect	OALD
bridge	1	1	1
burden	2		2
depict	2		3
emanate	1		1
leak	1		5
occupy	1		3
remark	1	1	4
retire	2	1	5
shed	1		2
troop	0		3

Table 8.3 Some subcategorization frames learned by Manning's system. For each verb, the table shows the number of correct and incorrect subcategorization frames that were learned and the number of frames listed in the Oxford Advanced Learner's Dictionary (Hornby 1974). Adapted from (Manning 1993).

"NP *in*-PP" as a subcategorization frame, and this was what was used as the gold standard for evaluation.

The third error in the table is the incorrect assignment of the intransitive frame to *remark*. This is probably due to sentences like (8.13) which look like *remark* is used without any arguments (except the subject).

(8.13) "And here we are 10 years later with the same problems," Mr. Smith remarked.

Recall in table 8.3 is relatively low. Recall here is the proportion of subcategorization frames listed in the OALD that were correctly identified. High precision and low recall are a consequence of the hypothesis testing framework adopted here. We only find subcategorization frames that are well attested. Conversely, this means that we do not find subcategorization frames that are rare. An example is the transitive use of *leak* as in *he leaked the news*, which was not found due to an insufficient number of occurrences in the corpus.

Table 8.3 is only a sample. Precision for the complete set of 40 verbs was 90%, recall was 43%. One way to improve these results would be to incorporate prior knowledge about a verb's subcategorization frame. While it is appealing to be able to learn just from raw data, without any help from a lexicographer's work, results will be much better if we take prior knowledge into account. The same pattern can be strong evidence for a new, unlisted subcategorization frame for one verb but evidence for a different frame with another verb. This is particularly true if we continue in the direction of more structured input to the subcategorization detector and use a parser instead of just a tagger. The simplest way of specifying prior knowledge would be to stipulate a higher prior for subcategorization frames listed in the dictionary.

As an example of how prior knowledge would improve accuracy, suppose we analyze a particular syntactic pattern (say, V NP S) and find two possible subcategorization frames f^1 (subject, object) and f^2 (subject, object, clause) with a slightly higher probability for f^1. This is our example (8.8). A parser could choose f^1 (subject, object) for a verb for which both frames have the same prior and f^2 (subject, object, clause) for a verb for which we have entered a bias against f^1 using some prior knowledge. For example, if we know that *email* is a verb of communication like *tell*, we may want to disfavor frames without clauses, and the parser would correctly choose frame f^2 (subject, object, clause) for *I emailed my boss where I had put the file with the slide presentation.* Such a system based on an incomplete subcategorization dictionary would make better use of a corpus than the systems described here and thus achieve better results.

Exercise 8.1 [⋆]

A potential problem with the inclusion of low-reliability cues is that they 'water down' the effectiveness of high-reliability cues if we combine all cues in one regular expression pattern, resulting in lower recall. How can we modify the hypothesis test to address this problem? Hint: Consider a multinomial distribution.

Exercise 8.2 [⋆]

Suppose a subcategorization frame for a verb is very rare. Discuss the difficulty of detecting such a frame with Brent and Manning's methods.

Exercise 8.3 [⋆]

Could one sharpen the hypothesis test for a low-frequency subcategorization frame f^j by taking as the event space the set of occurrences of the verb that could potentially be instances of the subcategorization frame? Consider a verb that is mostly used transitively (with a direct object NP), but that has some occurrences that subcategorize only for a PP. The methods discussed above would count transitive uses as evidence against the possibility of any intransitive use. With an appropriately reduced event space, this would no longer be true. Discuss advantages and disadvantages of such an approach.

Exercise 8.4 [⋆]

A difficult problem in an approach using a fixed significance level ($\alpha = 0.02$ in Brent's work) and a categorical classification scheme (the verb takes a particular frame, yes/no) is to determine the threshold such that as many subcategorization classifications as possible are correct (high precision), but not too many frames are missed (high recall). Discuss how this problem might be alleviated in a probabilistic framework in which we determine $P(f^j|v^i)$ instead of making a binary decision.

Exercise 8.5 [⋆]

In an approach to subcategorization acquisition based on parsing and priors, how would you combine probabilistic parses and priors into a posterior estimate of the probability of subcategorization frames? Assume that the priors are given in the form $P(f^j|v^i)$, and that parsing a corpus gives you a number of estimates of the form $P(s_k|f^j)$ (the probability of sentence k given that verb v^i in the sentence occurs with frame f^j).

8.3 Attachment Ambiguity

A pervasive problem in parsing natural language is resolving attachment ambiguities. When we try to determine the syntactic structure of a sentence, a problem that we consider in general in chapter 12, there are often phrases that can be attached to two or more different nodes in the tree, and we have to decide which one is correct. PP attachment is the attachment ambiguity problem that has received the most attention in the Statistical NLP literature. We saw an example of it in chapter 3 example (3.65), here repeated as (8.14):

(8.14) The children ate the cake with a spoon.

Depending on where we attach the prepositional phrase *with a spoon*, the sentence can either mean that the children were using a spoon to eat the cake (the PP is attached to *ate*), or that of the many cakes that they could have eaten the children ate the one that had a spoon attached (the PP is attached to *cake*). This latter reading is anomalous with this PP, but would be natural for the PP *with frosting*. See figure 3.2 in chapter 3 for the two different syntactic trees that correspond to the two attachments. This type of syntactic ambiguity occurs in every sentence in which a prepositional phrase follows an object noun phrase. The reason why the sentence in (1.12) had so many parses was because there were a lot of PPs (and participial relative clauses) which can attach at various places syntactically. In this section, we introduce a method for determining the attachment of *prepositional phrases* based on lexical information that is due to Hindle and Rooth (1993).

How are such ambiguities to be resolved? While one could imagine contextualizing a discourse where *with a spoon* was used as a differentiator of cakes, it was natural in the above example to see it as a tool for eating, and thus to choose the verb attachment. This seems to be true for many naturally occurring sentences:

(8.15) a. Moscow sent more than 100,000 soldiers into Afghanistan . . .

b. Sydney Water breached an agreement with NSW Health . . .

In these examples, only one attachment results in a reasonable interpretation. In (8.15a), the PP *into Afghanistan* must attach to the verb phrase headed by *send*, while in (8.15b), the PP *with NSW Health* must attach to the NP headed by *agreement*. In cases like these, lexical preferences can be used to disambiguate. Indeed, it turns out that, in most cases, simple lexical statistics can determine which attachment is the correct one. These simple statistics are basically co-occurrence counts between the verb and the preposition on the one hand, and between the noun and the preposition on the other. In a corpus, we would find lots of cases where *into* is used with *send*, but only a few where *into* is used with *soldier*. So we can be reasonably certain that the PP headed by *into* in (8.15a) attaches to *send*, not to *soldiers*.

A simple model based on this information is to compute the following likelihood ratio λ (cf. section 5.3.4 on likelihood ratios).

(8.16) $$\lambda(v, n, p) \quad = \quad \log \frac{P(p|v)}{P(p|n)}$$

where $P(p|v)$ is the probability of seeing a PP with p after the verb v and $P(p|n)$ is the probability of seeing a PP with p after the noun n. We can then attach to the verb for $\lambda(v, n, p) > 0$ and to the noun for $\lambda(v, n, p) < 0$.

The trouble with this model is that it ignores the fact that other things being equal, there is a preference for attaching phrases “low” in the parse tree. For PP attachment, the lower node is the NP node. For example, the tree in figure 3.2 (b) attaches the PP *with the spoon* to the lower NP node, the tree in figure 3.2 (a) attaches it to the higher VP node. One can explain low attachments with a preference for local operations. When we process the PP, the NP is still fresh in our mind and so it is easier to attach the PP to it.

w	$C(w)$	$C(w, with)$
end	5156	607
venture	1442	155

Table 8.4 An example where the simple model for resolving PP attachment ambiguity fails.

The following example from the *New York Times* shows why it is important to take the preference for attaching low into account:

(8.17) Chrysler confirmed that it would end its troubled venture with Maserati.

The preposition *with* occurs frequently after both *end* (e.g., *the show ended with a song*) and *venture* (e.g., *the venture with Maserati*). The data from the *New York Times* corpus in table 8.4,[4] when plugged into equation (8.16), predict attachment to the verb:

$$P(p|v) = \frac{607}{5156} \approx 0.118 > 0.107 \approx \frac{155}{1442} = P(p|n)$$

But that is the wrong decision here. The model is wrong because equation (8.16) ignores a bias for low attachment in cases where a preposition is equally compatible with the verb and the noun. We will now develop a probabilistic model for PP attachment that formalizes this bias.

8.3.1 Hindle and Rooth (1993)

In setting up the probabilistic model that is due to Hindle and Rooth (1993), we first define the event space. We are interested in sentences that are potentially ambiguous with respect to PP attachment. So we define the event space to consist of all clauses that have a transitive verb (a verb with an object noun phrase), an NP following the verb (the object noun phrase) and a PP following the NP.[5] Our goal is to resolve the PP attachment ambiguity in these cases.

In order to reduce the complexity of the model, we limit our attention to one preposition at a time (that is, we are not modeling possible interactions between PPs headed by different prepositions, see exercise 8.8),

[4] We used the subset of texts from chapter 5.

[5] Our terminology here is a little bit sloppy since the PP is actually part of the NP when it attaches to the noun, so, strictly speaking, it does not follow the NP. So what we mean here when we say “NP” is the base NP chunk without complements and adjuncts.

and, if there are two PPs with the same preposition in sequence, then we will only model the behavior of the first (see exercise 8.9).

To simplify the probabilistic model, we will not directly ask the question about whether a certain preposition is attached to a certain verb or noun. Rather, we will estimate how likely it is in general for a preposition to attach to a verb or noun. We will look at the following two questions, formalized by the sets of indicator random variables VA_p and NA_p:

VA_p: Is there a PP headed by p and following the verb v which attaches to v ($\mathrm{VA}_p = 1$) or not ($\mathrm{VA}_p = 0$)?

NA_p: Is there a PP headed by p and following the noun n which attaches to n ($\mathrm{NA}_p = 1$) or not ($\mathrm{NA}_p = 0$)?

Note that we are referring to any occurrence of the preposition p here rather than to a particular instance. So it is possible for both NA_p and VA_p to be 1 for some value of p. For instance, this is true for $p = on$ in the sentence:

(8.18) He put the book [*on* World War II] [*on* the table].

For a clause containing the sequence "$v \dots n \dots$PP," we wish to calculate the probability of the PP headed with preposition p attaching to the verb v and the noun n, conditioned on v and n:

$$P(\mathrm{VA}_p, \mathrm{NA}_p | v, n) = P(\mathrm{VA}_p | v, n) P(\mathrm{NA}_p | v, n) \tag{8.19}$$

$$= P(\mathrm{VA}_p | v) P(\mathrm{NA}_p | n) \tag{8.20}$$

In (8.19), we assume conditional independence of the two attachments – that is, whether a PP occurs modifying n is independent of whether one occurs modifying v. In (8.20), we assume that whether the verb is modified by a PP does not depend on the noun and whether the noun is modified by a PP does not depend on the verb.

That we are treating attachment of a preposition to a verb and to a noun (i.e., VA_p and NA_p) as independent events seems counterintuitive at first since the problem as stated above posits a binary choice between noun and verb attachment. So, rather than being independent, attachment to the verb seems to imply non-attachment to the noun and vice versa. But we already saw in (8.18) that the definitions of VA_p and NA_p imply that both can be true. The advantage of the independence assumption is that it is easier to derive empirical estimates for the two variables separately rather than estimating their joint distribution. We will see below how we can estimate the relevant quantities from an unlabeled corpus.

Now suppose that we wish to determine the attachment of a PP that is immediately following an object noun. We can compute an estimate in terms of model (8.20) by computing the probability of $\mathrm{NA}_p = 1$.

$$\begin{aligned} P(\mathrm{Attach}(p) = n|v, n) &= P(\mathrm{VA}_p = 0 \vee \mathrm{VA}_p = 1|v) \times P(\mathrm{NA}_p = 1|n) \\ &= 1.0 \times P(\mathrm{NA}_p = 1|n) \\ &= P(\mathrm{NA}_p = 1|n) \end{aligned}$$

So we do not need to consider whether $\mathrm{VA}_p = 0$ or $\mathrm{VA}_p = 1$, since while there could be other PPs in the sentence modifying the verb, they are immaterial to deciding the status of the PP immediately after the noun head.

In order to see that the case $\mathrm{VA}_p = 1$ and $\mathrm{NA}_p = 1$ does not make $\mathrm{Attach}(p) = v$ true, let's look at what these two premises entail. First, there must be two prepositional phrases headed by a preposition of type p. This is because we assume that any given PP can only attach to one phrase, either the verb or the noun. Second, the first of these two PPs must attach to the noun, the second to the verb. If it were the other way round, then we would get crossing brackets. It follows that $\mathrm{VA}_p = 1$ and $\mathrm{NA}_p = 1$ implies that the first PP headed by p is attached to the noun, not to the verb. So $\mathrm{Attach}(p) \neq v$ holds in this case.

In contrast, because there cannot be crossing lines in a phrase structure tree, in order for the first PP headed by the preposition p to attach to the verb, both $\mathrm{VA}_p = 1$ and $\mathrm{NA}_p = 0$ must hold. Substituting the appropriate values in model (8.20) we get:

$$\begin{aligned} P(\mathrm{Attach}(p) = v|v, n) &= P(\mathrm{VA}_p = 1, \mathrm{NA}_p = 0|v, n) \\ &= P(\mathrm{VA}_p = 1|v)P(\mathrm{NA}_p = 0|n) \end{aligned}$$

We can again assess $P(\mathrm{Attach}(p) = v)$ and $P(\mathrm{Attach}(p) = n)$ via a likelihood ratio λ.

$$\begin{aligned} \lambda(v, n, p) &= \log_2 \frac{P(\mathrm{Attach}(p) = v|v, n)}{P(\mathrm{Attach}(p) = n|v, n)} \\ &= \log_2 \frac{P(\mathrm{VA}_p = 1|v)P(\mathrm{NA}_p = 0|v)}{P(\mathrm{NA}_p = 1|n)} \end{aligned} \tag{8.21}$$

We choose verb attachment for large positive values of λ and noun attachment for large negative values. We can also make decisions for values of

λ closer to zero (verb attachment for positive λ and noun attachment for negative λ), but there is a higher probability of error.

How do we estimate the probabilities $P(\mathrm{VA}_p = 1|v)$ and $P(\mathrm{NA}_p = 1|n)$ that we need for equation (8.22)? The simplest method is to rely on maximum likelihood estimates of the familiar form:

$$\begin{aligned} P(\mathrm{VA}_p = 1|v) &= \frac{C(v,p)}{C(v)} \\ P(\mathrm{NA}_p = 1|n) &= \frac{C(n,p)}{C(n)} \end{aligned}$$

where $C(v)$ and $C(n)$ are the number of occurrences of v and n in the corpus, and $C(v,p)$ and $C(n,p)$ are the number of times that p attaches to v and p attaches to n. The remaining difficulty is to determine the attachment counts from an unlabeled corpus. In some sentences the attachment is obvious.

(8.22) a. The road *to London* is long and winding.

b. She sent him *into the nursery* to gather up his toys.

The prepositional phrase in italics in (8.22a) must attach to the noun since there is no preceding verb, and the italicized PP in (8.22b) must attach to the verb since attachment to a pronoun like *him* is not possible. So we can bump up our counts for $C(\mathit{road}, \mathit{to})$ and $C(\mathit{send}, \mathit{into})$ by one based on these two sentences. But many sentences are ambiguous. That, after all, is the reason why we need an automatic procedure for the resolution of attachment ambiguity.

Hindle and Rooth (1993) propose a heuristic for determining $C(v,p)$ and $C(n,p)$ from unlabeled data that has essentially three steps.

1. Build an initial model by counting all unambiguous cases (examples like (8.22a) and (8.22b)).

2. Apply the initial model to all ambiguous cases and assign them to the appropriate count if λ exceeds a threshold (for example, $\lambda > 2.0$ for verb attachment and $\lambda < -2.0$ for noun attachment).

3. Divide the remaining ambiguous cases evenly between the counts (that is, increase both $C(v,p)$ and $C(n,p)$ by 0.5 for each ambiguous case).

Sentence (8.15a), here repeated as (8.23), may serve as an example of how the method is applied (Hindle and Rooth 1993: 109–110).

(8.23) Moscow sent more than 100,000 soldiers into Afghanistan ...

First we estimate the two probabilities we need for the likelihood ratio. The count data are from Hindle and Rooth's test corpus.

$$\begin{aligned} P(\mathrm{VA}_{into} = 1 | send) &= \frac{C(send, into)}{C(send)} = \frac{86}{1742.5} \approx 0.049 \\ P(\mathrm{NA}_{into} = 1 | soldiers) &= \frac{C(soldiers, into)}{C(soldiers)} = \frac{1}{1478} \approx 0.0007 \end{aligned}$$

The fractional count is due to the step of the heuristic that divides the hardest ambiguous cases evenly between noun and verb. We also have:

(8.24) $P(\mathrm{NA}_{into} = 0 | soldiers) = 1 - P(\mathrm{NA}_{into} = 1 | soldiers) \approx 0.9993$

Plugging these numbers into formula (8.22), we get the following likelihood ratio.

$$\lambda(send, soldiers, into) \approx \log_2 \frac{0.049 \times 0.9993}{0.0007} \approx 6.13$$

So attachment to the verb is much more likely ($2^{6.13} \approx 70$ times more likely), which is the right prediction here. In general, the procedure is accurate in about 80% of cases if we always make a choice (Hindle and Rooth 1993: 115). We can trade higher precision for lower recall if we only make a decision for values of λ that exceed a certain threshold. For example, Hindle and Rooth (1993) found that precision was 91.7% and recall was 55.2% for $\lambda = 3.0$.

8.3.2 General remarks on PP attachment

Much of the early psycholinguistic literature on parsing emphasized the use of structural heuristics to resolve ambiguities, but they clearly don't help in cases like the PP attachments we have been looking at. For identical sequences of word classes, sometimes one parse structure is correct, and sometimes another. Rather, as suggested by Ford et al. (1982), lexical preferences seem very important here.

There are several major limitations to the model presented here. One is that it only considers the identity of the preposition and the noun and verb to which it might be attached. Sometimes other information is important (studies suggest human accuracy improves by around 5% when they see more than just a v, n, p triple). In particular, in sentences like those in (8.25), the identity of the noun that heads the NP inside the PP is clearly crucial:

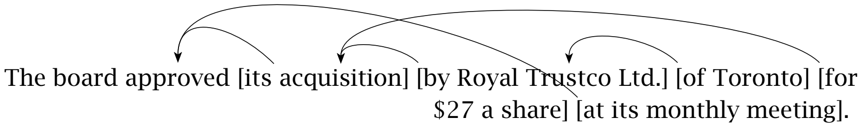
The board approved [its acquisition] [by Royal Trustco Ltd.] [of Toronto] [for
$27 a share] [at its monthly meeting].

Figure 8.2 Attachments in a complex sentence.

(8.25) a. I examined the man with a stethoscope

b. I examined the man with a broken leg

Other information might also be important. For instance Hindle and Rooth (1993) note that a superlative adjective preceding the noun highly biased things towards an NP attachment (in their data). This conditioning was probably omitted by Hindle and Rooth because of the infrequent occurrence of superlative adjectives. However, a virtue of the likelihood ratio approach is that other factors can be incorporated in a principled manner (providing that they are assumed to be independent). Much other work has used various other features, in particular the identity of the head noun inside the PP (Resnik and Hearst 1993; Brill and Resnik 1994; Ratnaparkhi et al. 1994; Zavrel et al. 1997; Ratnaparkhi 1998). Franz (1996) is able to include lots of features within a loglinear model approach, but at the cost of reducing the most basic association strength parameters to categorical variables.

A second major limitation is that Hindle and Rooth (1993) consider only the most basic case of a PP immediately after an NP object which is modifying either the immediately preceding noun or verb. But there are many more possibilities for PP attachments than this. Gibson and Pearlmutter (1994) argue that psycholinguistic studies have been greatly biased by their overconcentration on this one particular case. A PP separated from an object noun by another PP may modify any of the noun inside the preceding PP, the object noun, or the preceding verb. Figure 8.2 shows a variety of the distant and complex attachment patterns that occur in texts. Additionally, in a complex sentence, a PP might not modify just the immediately preceding verb, but might modify a higher verb. See Franz (1997) for further discussion, and exercise 8.9.

Other attachment issues

Apart from prepositional phrases, attachment ambiguity also occurs with various kinds of adverbial and participial phrases and clauses, and in *noun compounds*. The issue of the scope of coordinations in parsing is also rather similar to an attachment decision, but we will not consider it further here.

NOUN COMPOUNDS

A noun phrase consisting of a sequence of three or more nouns either has the left-branching structure [[N N] N] or the right-branching structure [N [N N]]. For example, *door bell manufacturer* is left-branching: *[[door bell] manufacturer]*. It's a manufacturer of door bells, not a manufacturer of bells that somehow has to do with doors. The phrase *woman aid worker* is an example of a right-branching NP: *[woman [aid worker]]*. The phrase refers to an aid worker who is female, not a worker working for or on *woman aid*. The left-branching case roughly corresponds to attachment of the PP to the verb ([V N P]), while the right-branching case corresponds to attachment to the noun ([V [N P]]).

We could directly apply the formalism we've developed for prepositional phrases to noun compounds. However, data sparseness tends to be a more serious problem for noun compounds than for prepositional phrases because prepositions are high-frequency words whereas most nouns are not. For this reason, one approach is to use some form of semantic generalization based on word classes in combination with attachment information. See Lauer (1995a) for one take on the problem (use of semantic classes for the PP attachment problem was explored by Resnik and Hearst (1993) with less apparent success). A different example of class-based generalization will be discussed in the next section.

As a final comment on attachment ambiguity, note that a large proportion of prepositional phrases exhibit 'indeterminacy' with respect to attachment (Hindle and Rooth 1993: 112). Consider the PP *with them* in (8.26):

(8.26) We have not signed a settlement agreement *with them*.

When you sign an agreement with person X, then in most cases it is an agreement with X, but you also do the signing with X. It is rather unclear whether the PP should be attached to the verb or the noun or whether we should rather say that a PP like *with them* in sentence (8.26) should attach to *both* verb and noun. Lauer (1995a) found that a significant proportion of noun compounds also had this type of attachment indeterminacy. This is an example of a possibly important insight that came out of Statistical NLP work. Before Hindle and Rooth's study, computational linguists were not generally aware of how widespread attachment indeterminacy is (though see Church and Patil (1982) for a counterexample).

After becoming aware of this fact, we could just say that it doesn't matter how we attach in indeterminate cases. But the phenomenon might also motivate us to explore new ways of determining the contribution a prepositional phrase makes to the meaning of a sentence. The phenomenon of attachment indeterminacy suggests that it may not be a good idea to require that PP meaning always be mediated through a noun phrase or a verb phrase as current syntactic formalisms do.

Exercise 8.6 [⋆]

As is usually the case with maximum likelihood estimates, they suffer in accuracy if data are sparse. Modify the estimation procedure using one of the procedures suggested in chapter 6. Hindle and Rooth (1993) use an 'Add One' method in their experiments.

Exercise 8.7 [⋆]

Hindle and Rooth (1993) used a partially parsed corpus to determine $C(v, p)$, and $C(n, p)$. Discuss whether we could use an unparsed corpus and what additional problems we would have to grapple with.

Exercise 8.8 [⋆]

Consider sentences with two PPs headed by two different prepositions, for example, "He put the book on Churchill in his backpack." The model we developed could attach *on Churchill* to *put* when applied to the preposition *on* and *in his backpack* to *book* when applied to the preposition *in*. But that is an incorrect parse tree since it has crossing brackets. Develop a model that makes consistent decisions for sentences with two PPs headed by different prepositions.

Exercise 8.9 [⋆⋆]

Develop a model that resolves the attachment of the second PP in a sequence of the form: V … N … PP PP. There are three possible cases here: attachment to the verb, attachment to the noun and attachment to the noun in the first PP.

Exercise 8.10 [⋆]

Note the following difference between a) the acquisition methods for attachment ambiguity in this section and b) those for subcategorization frames in the last section and those for collocations in chapter 5. In the case of PP attachment, we are interested in what is *predictable*. We choose the pattern that best fits what we would predict to happen from the training corpus. (For example, a PP headed by *in* after *send*.) In the case of subcategorization and collocations, we are interested in what is *unpredictable*, that is, patterns that shouldn't occur if our model was right. Discuss this difference.

8.4 Selectional Preferences

SELECTIONAL PREFERENCES SELECTIONAL RESTRICTIONS

Most verbs prefer arguments of a particular type. Such regularities are called *selectional preferences* or *selectional restrictions*. Examples are that the objects of the verb *eat* tend to be food items, the subjects of *think* tend to be people, and the subjects of *bark* tend to be dogs. These *semantic* constraints on arguments are analogous to the *syntactic* constraints we looked at earlier, subcategorization for objects, PPs, infinitives etc. We use the term *preferences* as opposed to *rules* because the preferences can be overridden in metaphors and other extended meanings. For example, *eat* takes non-food arguments in *eating one's words* or *fear eats the soul*.

The acquisition of selectional preferences is important in Statistical NLP for a number of reasons. If a word like *durian* is missing from our machine-readable dictionary, then we can infer part of its meaning from selectional restrictions. In the case of sentence (8.27), we can infer that a *durian* is a type of food.

(8.27) Susan had never eaten a fresh durian before.

Another important use of selectional preferences is for ranking the possible parses of a sentence. We will give higher scores to parses where the verb has 'natural' arguments than to those with atypical arguments, a strategy that allows us to choose among parses that are equally good on syntactic criteria. Scoring the semantic wellformedness of a sentence based on selectional preferences is more amenable to automated language processing than trying to understand the meaning of a sentence more fully. This is because the semantic regularities captured in selectional preferences are often quite strong and, due to the tight syntactic link between a verb and its arguments, can be acquired more easily from corpora than other types of semantic information and world knowledge.

We will now introduce the model of selectional preferences proposed by Resnik (1993, 1996). In principle, the model can be applied to any class of words that imposes semantic constraints on a grammatically dependent phrase: verb↔subject, verb↔direct object, verb↔prepositional phrase, adjective↔noun, noun↔noun (in noun-noun compounds). But we will only consider the case 'verb↔direct object' here, that is, the case of verbs selecting a semantically restricted class of direct object noun phrases.

The model formalizes selectional preferences using two notions: selec-

SELECTIONAL PREFERENCE STRENGTH

tional preference strength and selectional association. *Selectional preference strength* measures how strongly the verb constrains its direct object. It is defined as the KL divergence between the prior distribution of direct objects (the distribution of direct objects for verbs in general) and the distribution of direct objects of the verb we are trying to characterize.

We make two assumptions to simplify the model. First, we only take the *head noun* of the direct object into account (for example, *apple* in *Susan ate the green apple*) since the head is the crucial part of the noun phrase that determines compatibility with the verb. Second, instead of dealing with individual nouns, we will instead look at *classes* of nouns. As usual, a class-based model facilitates generalization and parameter estimation. With these assumptions, we can define selectional preference strength $S(v)$ as follows:

(8.28) $$S(v) = D(P(C|v)\|P(C)) = \sum_c P(c|v)\log\frac{P(c|v)}{P(c)}$$

where $P(C)$ is the overall probability distribution of noun classes and $P(C|v)$ is the probability distribution of noun classes in the direct object position of v. We can take the noun classes from any lexical resource that groups nouns into classes. Resnik (1996) uses WordNet.

SELECTIONAL ASSOCIATION

Based on selectional preference strength, we can define *selectional association* between a verb v and a class c as follows:

(8.29) $$A(v,c) = \frac{P(c|v)\log\frac{P(c|v)}{P(c)}}{S(v)}$$

That is, the association between a verb and a class is defined as the proportion that its summand $P(c|v)\log\frac{P(c|v)}{P(c)}$ contributes to the overall preference strength $S(v)$.

Finally, we need a rule for assigning association strength to nouns (as opposed to noun classes). If the noun n is in only one class c, then we simply define $A(v,n) \overset{\text{def}}{=} A(v,c)$. If the noun is a member of several classes, then we define its association strength as the highest association strength of any of its classes.

(8.30) $$A(v,n) = \max_{c\in\text{classes}(n)} A(v,c)$$

A noun like *chair* in (8.31) is in several classes because it is polysemous (or ambiguous).

(8.31) Susan interrupted the chair.

Noun class c	$P(c)$	$P(c\|eat)$	$P(c\|see)$	$P(c\|find)$
people	0.25	0.01	0.25	0.33
furniture	0.25	0.01	0.25	0.33
food	0.25	0.97	0.25	0.33
action	0.25	0.01	0.25	0.01
SPS $S(v)$		1.76	0.00	0.35

Table 8.5 Selectional Preference Strength (SPS). The argument distributions and selectional preference strengths of three verbs for a classification of nouns with four classes (based on hypothetical data).

DISAMBIGUATION

In the case of *chair*, we have two candidate classes, 'furniture' and 'people' (the latter in the sense 'chairperson'). Equating $A(v, n)$ with the maximum $A(v, c)$ amounts to disambiguating the noun. In sentence (8.31) we will base the association strength $A(\textit{interrupt}, \textit{chair})$ on the class 'people' since interrupting people is much more common than interrupting pieces of furniture, that is:

$$A(\textit{interrupt}, \text{people}) \gg A(\textit{interrupt}, \text{furniture})$$

Hence:

$$\begin{aligned} A(\textit{interrupt}, \textit{chair}) &= \max_{c \in \text{classes}(\textit{chair})} A(\textit{interrupt}, c) \\ &= A(\textit{interrupt}, \text{people}) \end{aligned}$$

So we can disambiguate *chair* as a by-product of determining the association of *interrupt* and *chair*.

The hypothetical data in table 8.5 (based on (Resnik 1996: 139)) may serve as a further illustration of the model. The table shows the prior distribution of object NPs over noun classes (assuming that there are only the four classes shown) and posterior distributions for three verbs. The verb *eat* overwhelmingly prefers food items as arguments; *see*'s distribution is not very different from the prior distribution since all physical objects can be seen; *find* has a uniform distribution over the first three classes, but 'disprefers' actions since actions are not really the type of entities that are found.

The selectional preference strengths of the three verbs are shown in the row 'SPS.' The numbers conform well with our intuition about the three verbs: *eat* is very specific with respect to the arguments it can take, *find* is less specific, and *see* has no selectional preferences (at least in our hypothetical data). Note that there is a clear interpretation of SPS as the amount of information we gain about the argument after learning about the verb. In the case of *eat*, SPS is 1.76, corresponding to almost 2 binary questions. That is just the number of binary questions we need to get from four classes (people, furniture, food, action) to one, namely the class 'food' that *eat* selects. (Binary logarithms were used to compute SPS and association strength.)

Computing the association strengths between verbs and noun classes, we find that the class 'food' is strongly preferred by *eat* (8.32) whereas the class 'action' is dispreferred by *find* (8.33). This example shows that the model formalizes selectional 'dispreferences' (negative numbers) as well as selectional preferences (positive numbers).

(8.32) $A(\mathit{eat}, \text{food}) = 1.08$

(8.33) $A(\mathit{find}, \text{action}) = -0.13$

The association strengths between *see* and all four noun classes are zero, corresponding to the intuition that *see* does not put strong constraints on its possible arguments.

The remaining problem is to estimate the probability that a direct object in noun class c occurs given a verb v, $P(c|v) = \frac{P(v,c)}{P(v)}$. The maximum likelihood estimate for $P(v)$ is $C(v)/\sum_{v'} C(v')$, the relative frequency of v with respect to all verbs. Resnik (1996) proposes the following estimate for $P(v,c)$:

(8.34) $$P(v,c) = \frac{1}{N} \sum_{n \in \text{words}(c)} \frac{1}{|\text{classes}(n)|} C(v,n)$$

where N is the total number of verb-object pairs in the corpus, $\text{words}(c)$ is the set of all nouns in class c, $\text{classes}(n)$ is the number of noun classes that contain n as a member and $C(v,n)$ is the number of verb-object pairs with v as the verb and n as the head of the object NP. This way of estimating $P(v,c)$ bypasses the problem of disambiguating nouns. If a noun that is a member of two classes c_1 and c_2 occurs with v, then we assign half of this occurrence to $P(v,c_1)$ and half to $P(v,c_2)$.

So far, we have only presented constructed examples. Table 8.6 shows some actual data from Resnik's experiments on the Brown corpus (Resnik 1996: 142). The verbs and nouns were taken from a psycholinguistic study (Holmes et al. 1989). The nouns in the left and right halves of the table are 'typical' and 'atypical' objects, respectively. For most verbs,

Verb v	Noun n	$A(v,n)$	Class	Noun n	$A(v,n)$	Class
answer	*request*	4.49	speech act	*tragedy*	3.88	communication
find	*label*	1.10	abstraction	*fever*	0.22	psych. feature
hear	*story*	1.89	communication	*issue*	1.89	communication
remember	*reply*	1.31	statement	*smoke*	0.20	article of commerce
repeat	*comment*	1.23	communication	*journal*	1.23	communication
read	*article*	6.80	writing	*fashion*	−0.20	activity
see	*friend*	5.79	entity	*method*	−0.01	method
write	*letter*	7.26	writing	*market*	0.00	commerce

Table 8.6 Association strength distinguishes a verb's plausible and implausible objects. The left half of the table shows typical objects, the right half shows atypical objects. In most cases, association strength $A(v,n)$ is a good predictor of object typicality.

association strength accurately predicts which object is typical. For example, it correctly predicts that *friend* is a more natural object for *see* than *method*. Most errors the model makes are due to the fact that it performs a form of disambiguation, by choosing the highest association strength among the possible classes of the noun (cf. the example of *chair* we discussed earlier). Even if a noun is an atypical object, if it has a rare interpretation as a plausible object, then it will be rated as typical. An example of this is *hear*. Both *story* and *issue* can be forms of communication, but this meaning is rarer for *issue*. Yet the model chooses the rare interpretation because it makes more sense for the verb *hear*.

IMPLICIT OBJECT ALTERNATION

Apart from the specific question of selectional preference, Resnik also investigates how well the model predicts whether or not a verb has the so-called *implicit object alternation* (or *unspecified object alternation*, see Levin (1993: 33)). An example is the alternation between sentences (8.35a) and (8.35b). The verb *eat* alternates between explicitly naming what was eaten (8.35a) and leaving the thing eaten implicit (8.35b).

(8.35) a. Mike ate the cake.

b. Mike ate.

The explanation Resnik offers for this phenomenon is that the more constraints a verb puts on its object, the more likely it is to permit the implicit-object construction. The intuition is that for a verb like *eat* with a strong selectional preference, just knowing the verb gives us so much information about the direct object that we don't have to mention it. Resnik finds evidence that selectional preference strength is a good predictor of the permissibility of the implicit-object alternation for verbs.

We can now see why Resnik's model defines selectional preference strength (SPS) as the primary concept and derives association strength from it. SPS is seen as the more basic phenomenon which explains the occurrence of implicit objects as well as association strength.

An alternative is to define association strength directly as $P(c|v)$ – or as $P(n|v)$ if we don't want to go through an intermediate class representation. Approaches to computing $P(n|v)$ include distributional clustering (the work by Pereira et al. (1993) described in chapter 14) and methods for computing the similarity of nouns. If a measure of the similarity of nouns is available, then $P(n|v)$ can be computed from the distribution of nouns similar to n that are found in the argument slot of v. See the next section for more on this approach.

Exercise 8.11 [⋆]

As we pointed out above, we can use a model of selectional preferences for disambiguating nouns by computing the association strengths for different senses of the noun. This strategy assumes that we know what the senses of the noun are and which classes they are members of. How could one use selectional preferences to discover senses of nouns whose senses we don't know?

Exercise 8.12 [⋆]

Verbs can also be ambiguous as in the case of *fire* in these two sentences.

(8.36) a. The president fired the chief financial officer.

b. Mary fired her gun first.

How can the model be used to disambiguate verbs? Consider two scenarios, one in which we have a training set in which verb senses are labeled, one in which we don't.

Exercise 8.13 [⋆]

The model discussed in this section assigns the noun sense with the maximum association strength. This approach does not take prior probabilities into account. We may not want to choose an extremely rare sense of a noun even if it is the best fit as the object NP of a verb.

Example: The noun *shot* has the rare meaning 'marksman' as in *John was reputed to be a crack shot.* So, theoretically, we could choose this sense for *shot* in the sentence *John fired a shot*, corresponding to the meaning *John laid off a marksman.*

How could prior probabilities be used to avoid such incorrect inferences?

Exercise 8.14 [⋆⋆]

In the approach developed above, WordNet is treated as a flat set of noun classes, but it is actually a hierarchy. How could one make use of the information present in the hierarchy (for example, the fact, that the class 'dog' is a subclass of 'animal' which in turn is a subclass of 'entity')?

Exercise 8.15 [⋆⋆]

Verbs can be organized into a hierarchy too. How could one use hierarchical information about verbs for better parameter estimation?

Exercise 8.16 [⋆]

One assumption of the model is that it is the head noun that determines the compatibility of an object NP with the selectional preferences of the verb. However, as pointed out by Resnik (1996: 137), that is not always the case. Examples include negation (*you can't eat stones*), and certain adjectival modifiers (*he ate a chocolate firetruck*; *the tractor beam pulled the ship closer*); neither stones nor firetrucks are compatible with the selectional preferences of *eat*, but these sentences are still well-formed. Discuss this problem.

Exercise 8.17 [⋆]

Hindle and Rooth (1993) go through several iterations of estimating initial parameters of their model, disambiguating some ambiguous attachments and reestimating parameters based on disambiguated instances. How could this approach be used to estimate the prior probabilities of noun classes in (8.34)? The goal would be to improve on the uniform distribution over possible classes assumed in the equation.

Exercise 8.18 [⋆]

Resnik's model expresses association strength as a proportion of selectional preference strength. This leads to interesting differences from an approach based on formalizing selectional preference as $P(n|v)$. Compare two noun-verb pairs with equal $P(n|v)$, that is, $P(n_1|v_1) = P(n_2|v_2)$. If the selectional preference strength of v_1 is much larger than that of v_2, then we get $A(v_1, c(n_1)) \ll A(v_2, c(n_2))$. So the two models make different predictions here. Discuss these differences.

8.5 Semantic Similarity

The holy grail of lexical acquisition is the acquisition of meaning. There are many tasks (like text understanding and information retrieval) for which Statistical NLP could make a big difference if we could automatically acquire meaning. Unfortunately, how to represent meaning in a way that can be operationally used by an automatic system is a largely unsolved problem. Most work on acquiring semantic properties of words

SEMANTIC SIMILARITY

has therefore focused on *semantic similarity*. Automatically acquiring a relative measure of how similar a new word is to known words (or how dissimilar) is much easier than determining what the meaning actually is.

GENERALIZATION

Despite its limitations, semantic similarity is still a useful measure to have. It is most often used for *generalization* under the assumption that semantically similar words behave similarly. An example would be the problem of selectional preferences that we discussed in the previous section. Suppose we want to find out how appropriate *durian* is as an argument of *eat* in sentence (8.37) (our previous example (8.27)):

(8.37) Susan had never eaten a fresh durian before.

Suppose further that we don't have any information about *durian* except that it's semantically similar to *apple*, *banana*, and *mango*, all of which perfectly fit the selectional preferences of *eat*. Then we can generalize from the behavior of *apple*, *banana*, and *mango* to the semantically similar *durian* and hypothesize that *durian* is also a good argument of *eat*. This scheme can be implemented in various ways. We could base our treatment of *durian* only on the closest semantic neighbor (say, *mango*), or we could base it on a combination of evidence from a fixed number of nearest neighbors, a combination that can be weighted according to how semantically similar each neighbor is to *durian*.

CLASS-BASED GENERALIZATION

Similarity-based generalization is a close relative of class-based generalization. In similarity-based generalization we only consider the closest neighbors in generalizing to the word of interest. In class-based generalization, we consider the whole class of elements that the word of interest is most likely to be a member of. (See exercise 8.20.)

Semantic similarity is also used for query expansion in information retrieval. A user who describes a request for information in her own words may not be aware of related terms which are used in the documents that the user would be most interested in. If a user describes a request for documents on Russian space misions using the word *astronaut*, then a query expansion system can suggest the term *cosmonaut* based on the semantic similarity between *astronaut* and *cosmonaut*.

k NEAREST NEIGHBORS
KNN

Another use of semantic similarity is for so-called *k nearest neighbors* (or *KNN*) classification, see section 16.4). We first need a training set of elements that are each assigned to a category. The elements might be words and the categories might be topic categories as they are used by newswire services ('financial,' 'agriculture,' 'politics' etc.). In KNN classification we assign a new element to the category that is most prevalent among its k nearest neighbors.

Before delving into the details of how to acquire measures of semantic similarity, let us remark that semantic similarity is not as intuitive and clear a notion as it may seem at first. For some, semantic similarity is an extension of synonymy and refers to cases of near-synonymy like the pair *dwelling/abode*. Often semantic similarity refers to the notion that two words are from the same *semantic domain* or *topic*. On this understanding of the term, words are similar if they refer to entities in the world that are likely to co-occur like *doctor*, *nurse*, *fever*, and *intravenous*, words that can refer to quite different entities or even be members of different syntactic categories.

SEMANTIC DOMAIN
TOPIC

One attempt to put the notion of semantic similarity on a more solid footing is provided by Miller and Charles (1991), who show that judgements of semantic similarity can be explained by the degree of *contextual interchangeability* or the degree to which one word can be substituted for another in context.

CONTEXTUAL INTERCHANGEABILITY

Note that ambiguity presents a problem for all notions of semantic similarity. If a word is semantically similar to one sense of an ambiguous word, then it is rarely semantically similar to the other sense. For example, *litigation* is similar to the legal sense of *suit*, but not to the 'clothes' sense. When applied to ambiguous words, semantically similar usually means 'similar to the appropriate sense'.

8.5.1 Vector space measures

A large class of measures of semantic similarity are best conceptualized as measures of vector similarity. The two words whose semantic similarity we want to compute are represented as vectors in a multi-dimensional space. Figures 8.3, 8.4, and 8.5 give (constructed) examples of such multidimensional spaces (see also figure 15.5).

DOCUMENT SPACE

The matrix in figure 8.3 represents words as vectors in *document space*. Entry a_{ij} contains the number of times word j occurs in document i. Words are deemed similar to the extent that they occur in the same documents. In document space, *cosmonaut* and *astronaut* are dissimilar (no shared documents); *truck* and *car* are similar since they share a document: they co-occur in d_4.

WORD SPACE

The matrix in figure 8.4 represents words as vectors in *word space*. Entry b_{ij} contains the number of times word j co-occurs with word i.

	cosmonaut	astronaut	moon	car	truck
d_1	1	0	1	1	0
d_2	0	1	1	0	0
d_3	1	0	0	0	0
d_4	0	0	0	1	1
d_5	0	0	0	1	0
d_6	0	0	0	0	1

Figure 8.3 A document-by-word matrix A.

	cosmonaut	astronaut	moon	car	truck
cosmonaut	2	0	1	1	0
astronaut	0	1	1	0	0
moon	1	1	2	1	0
car	1	0	1	3	1
truck	0	0	0	1	2

Figure 8.4 A word-by-word matrix B.

	cosmonaut	astronaut	moon	car	truck
Soviet	1	0	0	1	1
American	0	1	0	1	1
spacewalking	1	1	0	0	0
red	0	0	0	1	1
full	0	0	1	0	0
old	0	0	0	1	1

Figure 8.5 A modifier-by-head matrix C. The nouns (or heads of noun phrases) in the top row are modified by the adjectives in the left column.

Co-occurrence can be defined with respect to documents, paragraphs or other units. Words are similar to the extent that they co-occur with the same words. Here, *cosmonaut* and *astronaut* are more similar than before since they both co-occur with *moon*.

We have defined co-occurrence in figure 8.4 with respect to the documents in figure 8.3. In other words, the following relationship holds: $B = A^T A$. (Here $\cdot^T$ is the *transpose*, where we swap the rows and columns so that $X^T_{ij} = X_{ji}$.)

TRANSPOSE

The matrix in figure 8.5 represents nouns (interpreted as heads of noun

MODIFIER SPACE

phrases) as vectors in *modifier space*. Entry c_{ij} contains the number of times that head j is modified by modifier i. Heads are similar to the extent that they are modified by the same modifiers. Again, *cosmonaut* and *astronaut* are similar. But, interestingly *moon* is dissimilar from *cosmonaut* and *astronaut* here, in contrast to the document space in figure 8.3 and the word space in figure 8.4. This contrast demonstrates that different spaces get at different types of semantic similarity. The type of undifferentiated co-occurrence information in document and word spaces captures *topical similarity* (words pertaining to the same topic domain). Head-modifier information is more fine-grained. Although *astronaut* and *moon* are part of the same domain ('space exploration'), they are obviously entities with very different properties (a human being versus a celestial body). Different properties correspond to different modifiers, which explains why the two words come out as dissimilar on the head-modifier metric.[6]

TOPICAL SIMILARITY

The three matrices also have an interesting interpretation if we look at the similarity of *rows* instead of the similarity of *columns* (or, equivalently, look at the similarity of columns of the transposed matrices). Looking at the matrices this way, A defines similarity between documents. This is the standard way of defining similarity among documents and between documents and queries in information retrieval. Matrix C defines similarity between modifiers when transposed. For example, *red* and *old* are similar (they share *car* and *truck*), suggesting that they are used to modify the same types of nouns. Matrix B is *symmetric*, so looking at similarity of rows is no different from looking at similarity of columns.

BINARY VECTORS

So far we have appealed to an intuitive notion of vector similarity. Table 8.7 defines several measures that have been proposed to make this notion precise (adapted from (van Rijsbergen 1979: 39)). At first, we only consider *binary vectors*, that is, vectors with entries that are either 0 or 1. The simplest way to describe a binary vector is as the set of dimensions on which it has non-zero values. So, for example, the vector for *cosmonaut* in figure 8.5 can be represented as the set $\{$*Soviet*, *spacewalking*$\}$. Having done this, we can calculate similarities using set operations, as in table 8.7.

[6] See Grefenstette (1996) and Schütze and Pedersen (1997) for a discussion of the pros and cons of measuring word similarity based on associations versus head-modifier relationships.

Similarity measure	Definition
matching coefficient	$X \cap Y$
Dice coefficient	$\frac{2\|X \cap Y\|}{\|X\|+\|Y\|}$
Jaccard (or Tanimoto) coefficient	$\frac{\|X \cap Y\|}{\|X \cup Y\|}$
Overlap coefficient	$\frac{\|X \cap Y\|}{\min(\|X\|,\|Y\|)}$
cosine	$\frac{\|X \cap Y\|}{\sqrt{\|X\| \times \|Y\|}}$

Table 8.7 Similarity measures for binary vectors.

MATCHING COEFFICIENT

The first similarity measure, the *matching coefficient*, simply counts the number of dimensions on which both vectors are non-zero. In contrast to the other measures, it does not take into account the length of the vectors and the total number of non-zero entries in each.[7]

DICE COEFFICIENT

The *Dice coefficient* normalizes for length by dividing by the total number of non-zero entries. We multiply by 2 so that we get a measure that ranges from 0.0 to 1.0 with 1.0 indicating identical vectors.

JACCARD COEFFICIENT

The *Jaccard coefficient* penalizes a small number of shared entries (as a proportion of all non-zero entries) more than the Dice coefficient does. Both measures range from 0.0 (no overlap) to 1.0 (perfect overlap), but the Jaccard coefficient gives lower values to low-overlap cases. For example, two vectors with ten non-zero entries and one common entry get a Dice score of $2 \times 1/(10+10) = 0.1$ and a Jaccard score of $1/(10+10-1) \approx 0.05$. The Jaccard coefficient is frequently used in chemistry as a measure of similarity between chemical compounds (Willett and Winterman 1986).

OVERLAP COEFFICIENT

The *Overlap coefficient* has the flavor of a measure of inclusion. It has a value of 1.0 if every dimension with a non-zero value for the first vector is also non-zero for the second vector or vice versa (in other words if $X \subseteq Y$ or $Y \subseteq X$).

COSINE

The *cosine* is identical to the Dice coefficient for vectors with the same number of non-zero entries (see exercise 8.24), but it penalizes less in cases where the number of non-zero entries is very different. For example, if we compare one vector with one non-zero entry and another vector with 1000 non-zero entries and if there is one shared entry, then

[7] This can be desirable to reflect our confidence in the similarity judgement. Hindle (1990) recommends a measure for noun similarity with this property.

we get a Dice coefficient of $2 \times 1/(1+1000) \approx 0.002$ and a cosine of $1/\sqrt{1000 \times 1} \approx 0.03$. This property of the cosine is important in Statistical NLP since we often compare words or objects that we have different amounts of data for, but we don't want to say they are dissimilar just because of that.

VECTOR SPACE

So far we have looked at binary vectors, but binary vectors only have one bit of information on each dimension. A more powerful representation for linguistic objects is the real-valued *vector space*. We will not give a systematic introduction to linear algebra here, but let us briefly review the basic concepts of vector spaces that we need in this book. A real-valued vector $\vec{x}$ of dimensionality n is a sequence of n real numbers, where x_i denotes the i^{th} component of $\vec{x}$ (its value on dimension i). The components of a vector are properly written as a column:

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \tag{8.38}$$

LENGTH OF A VECTOR

However, we sometimes write vectors horizontally within paragraphs. We write $\mathbb{R}^n$ for the vector space of real-valued vectors with dimensionality n, so we have $\vec{x} \in \mathbb{R}^n$. In a Euclidean vector space, the *length of a vector* is defined as follows.

$$|\vec{x}| = \sqrt{\sum_{i=1}^{n} x_i^2} \tag{8.39}$$

Finally, the dot product between two vectors is defined as $\vec{x} \cdot \vec{y} = \sum_{i=1}^{n} x_i y_i$.

The cosine, the last similarity measure we introduced for binary vectors, is also the most important one for real-valued vectors. The cosine measures the cosine of the angle between two vectors. It ranges from 1.0 ($\cos(0^\circ) = 1.0$) for vectors pointing in the same direction over 0.0 for orthogonal vectors ($\cos(90^\circ) = 0.0$) to -1.0 for vectors pointing in opposite directions ($\cos(180^\circ) = -1.0$).

For the general case of two n-dimensional vectors $\vec{x}$ and $\vec{y}$ in a real-valued space, the cosine measure can be calculated as follows:

$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}||\vec{y}|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \tag{8.40}$$

NORMALIZED CORRELATION COEFFICIENT

This definition highlights another interpretation of the cosine, the interpretation as the *normalized correlation coefficient*. We compute how well the x_i and the y_i correlate and then divide by the (Euclidean) length of the two vectors to scale for the magnitude of the individual x_i and y_i.

NORMALIZATION

We call a vector *normalized* if it has unit length according to the Euclidean norm:

$$|x| = \sum_{i=1}^{n} x_i^2 = 1 \tag{8.41}$$

For normalized vectors, the cosine is simply the dot product:

$$\cos(\vec{x}, \vec{y}) = \vec{x} \cdot \vec{y} \tag{8.42}$$

EUCLIDEAN DISTANCE

The *Euclidean distance* between two vectors measures how far apart they are in the vector space:

$$|\vec{x} - \vec{y}| = \sqrt{\textstyle\sum_{i=1}^{n} (x_i - y_i)^2} \tag{8.43}$$

An interesting property of the cosine is that, if applied to normalized vectors, it will give the same ranking of similarities as Euclidean distance does. That is, if we only want to know which of two objects is closest to a third object, then cosine and Euclidean distance give the same answer for normalized vectors. The following derivation shows why ranking according to cosine and Euclidean distance comes out to be the same:

$$\begin{aligned}(|\vec{x} - \vec{y}|)^2 &= \sum_{i=1}^{n} (x_i - y_i)^2 \\ &= \sum_{i=1}^{n} x_i^2 - 2\sum_{i=1}^{n} x_i y_i + \sum_{i=1}^{n} y_i^2 \\ &= 1 - 2\sum_{i=1}^{n} x_i y_i + 1 \\ &= 2(1 - \vec{x} \cdot \vec{y})\end{aligned} \tag{8.44}$$

Finally, the cosine has also been used as a similarity measure of probability distributions (Goldszmidt and Sahami 1998). Two distributions $\{p_i\}$ and $\{q_i\}$ are first transformed into $\{\sqrt{p_i}\}$ and $\{\sqrt{q_i}\}$. Taking the cosine of the two resulting vectors gives the measure $D = \sum_{i=1}^{n} \sqrt{p_i q_i}$, which can be interpreted as the sum over the geometric means of the $\{p_i\}$ and $\{q_i\}$.

Table 8.8 shows some cosine similarities computed for the *New York Times* corpus described in chapter 5. We compiled a 20,000-by-1,000 matrix similar to the word-by-word matrix in figure 8.4. As rows we selected

Focus word	Nearest neighbors							
garlic	*sauce*	.732	*pepper*	.728	*salt*	.726	*cup*	.726
fallen	*fell*	.932	*decline*	.931	*rise*	.930	*drop*	.929
engineered	*genetically*	.758	*drugs*	.688	*research*	.687	*drug*	.685
Alfred	*named*	.814	*Robert*	.809	*William*	.808	*W*	.808
simple	*something*	.964	*things*	.963	*You*	.963	*always*	.962

Table 8.8 The cosine as a measure of semantic similarity. For each of the five words in the left column, the table shows the words that were most similar according to the cosine measure when applied to a word-by-word co-occurrence matrix. For example, *sauce* is the word that is most similar to *garlic*. The cosine between the vectors of *sauce* and *garlic* is 0.732.

the 20,000 most frequent words, as columns the 1,000 most frequent words (after elimination of the 100 most frequent words in both cases). Instead of raw co-occurrence counts, we used the logarithmic weighting function $f(x) = 1 + \log(x)$ for non-zero counts (see section 15.2.2). A co-occurrence event was defined as two words occurring within 25 words of each other. The table shows cosine similarities between rows of the matrix.

For some word pairs, cosine in word space is a good measure of semantic similarity. The neighbors of *garlic* are generally close in meaning to garlic (with the possible exception of *cup*). The same is true for *fallen*. Note, however, that grammatical distinctions are not reflected because co-occurrence information is insensitive to word order and grammatical dependencies (the past participle *fallen* and the past tense *fell* are nearest neighbors of each other). The word *engineered* shows the corpus-dependency of the similarity measure. In the *New York Times*, the word is often used in the context of genetic engineering. A corpus of automobile magazine articles would give us a very different set of neighbors of *engineered*. Finally, the words *Alfred* and *simple* show us the limits of the chosen similarity measure. Some of the neighbors of *Alfred* are also names, but this is a case of part-of-speech similarity rather than semantic similarity. The neighbors of *simple* seem completely random. Since *simple* is frequently used and its occurrences are distributed throughout the corpus, co-occurrence information is not useful here to characterize the semantics of the word.

The examples we have given demonstrate the advantage of vector spaces as a representational medium: their simplicity. It is easy to visualize vectors in a two-dimensional or three-dimensional space. Equating similarity with the extent to which the vectors point in the same direction is equally intuitive. In addition, vector space measures are easy to compute. Intuitive simplicity and computational efficiency are probably the main reasons that vector space measures have been used for a long time in information retrieval, notably for word-by-document matrices (Lesk 1969; Salton 1971a; Qiu and Frei 1993). Work on using vector measures for word-by-word and modifier-by-head matrices is more recent (Grefenstette 1992b; Schütze 1992b). See (Grefenstette 1992a) and (Burgess and Lund 1997) for research demonstrating that vector-based similarity measures correspond to psychological notions of semantic similarity such as the degree to which one word *primes* another. PRIMING

8.5.2 Probabilistic measures

The problem with vector space based measures is that, except for the cosine, they operate on binary data (yes or no). The cosine is the only vector space measure that accommodates quantitative information, but it has its own problems. Computing the cosine assumes a Euclidean space. This is because the cosine is defined as the ratio of the lengths of two sides of a triangle. So we need a measure of length, the Euclidean metric. But a Euclidean space is not a well-motivated choice if the vectors we are dealing with are vectors of probabilities or counts – which is what most representations for computing semantic similarity are based on. To see this observe that the Euclidean distance between the probabilities 0.0 and 0.1 is the same as the distance between the probabilities 0.9 and 1.0. But in the first case we have the difference between impossibility and a chance of 1 in 10 whereas in the second there is only a small difference of about 10%. The Euclidean distance is appropriate for normally distributed quantities, not for counts and probabilities.

Matrices of counts like those in figures 8.3, 8.4, and 8.5 can be easily transformed into matrices of conditional probabilities by dividing each element in a row by the sum of all entries in the row (this amounts to using maximum likelihood estimates). For example, in the matrix in figure 8.5, the entry for $(\mathit{American}, \mathit{astronaut})$ would be transformed into $P(\mathit{American}|\mathit{astronaut}) = \frac{1}{2} = 0.5$. The question of semantic similarity can then be recast as a question about the similarity (or dissimilarity) of two probability distributions.

(Dis-)similarity measure	Definition
KL divergence	$D(p\|q) = \sum_i p_i \log \frac{p_i}{q_i}$
information radius (IRad)	$D(p\|\frac{p+q}{2}) + D(q\|\frac{p+q}{2})$
L_1 norm	$\sum_i \lvert p_i - q_i \rvert$

Table 8.9 Measures of (dis-)similarity between probability distributions.

Table 8.9 shows three measures of dissimilarity between probability distributions investigated by Dagan et al. (1997b). We are already familiar with the KL divergence from section 2.2.5. It measures how well distribution q approximates distribution p; or, more precisely, how much information is lost if we assume distribution q when the true distribution is p. The KL divergence has two problems for practical applications. First, we get a value of ∞ if there is a 'dimension' with $q_i = 0$ and $p_i \neq 0$ (which will happen often, especially if we use simple maximum likelihood estimates). Secondly, KL divergence is asymmetric, that is, usually $D(p\|q) \neq D(q\|p)$. The intuitive notion of semantic similarity and most other types of similarity we are interested in is symmetric, so the following should hold: $\text{sim}(p, q) = \text{sim}(q, p)$.[8]

KL DIVERGENCE

INFORMATION RADIUS

The second measure in table 8.9, *information radius* (or total divergence to the average as Dagan et al. (1997b) call it), overcomes both these problems. It is symmetric ($\text{IRad}(p, q) = \text{IRad}(q, p)$) and there is no problem with infinite values since $\frac{p_i+q_i}{2} \neq 0$ if either $p_i \neq 0$ or $q_i \neq 0$. The intuitive interpretation of IRad is that it answers the question: How much information is lost if we describe the two words (or random variables in the general case) that correspond to p and q with their average distribution? IRad ranges from 0 for identical distributions to $2 \log 2$ for maximally different distributions (see exercise 8.26). As usual we assume $0 \log 0 = 0$.

L_1 NORM

MANHATTAN NORM

A third measure considered by Dagan et al. (1997b) is the L_1 (or *Manhattan*) *norm*. It also has the desirable properties of being symmetric and well-defined for arbitrary p and q. We can interpret it as a measure of *the expected proportion of different events*, that is, as the expected propor-

[8] Note that in clustering, asymmetry can make sense since we are comparing two different entities, the individual word that we need to assign to a cluster and the representation of the cluster. The question here is how well the cluster represents the word which is different from similarity in the strict sense of the word. See (Pereira et al. 1993).

tion of events that are going to be different between the distributions p and q. This is because $\frac{1}{2}L_1(p,q) = 1 - \sum_i \min(p_i, q_i)$, and $\sum_i \min(p_i, q_i)$ is the expected proportion of trials with the same outcome.[9]

As an example consider the following conditional distributions computed from the data in figure 8.5.

$$\begin{aligned}
p_1 &= P(\mathit{Soviet}|\mathit{cosmonaut}) = 0.5 \\
p_2 &= 0 \\
p_3 &= P(\mathit{spacewalking}|\mathit{cosmonaut}) = 0.5 \\
q_1 &= 0 \\
q_2 &= P(\mathit{American}|\mathit{astronaut}) = 0.5 \\
q_3 &= P(\mathit{spacewalking}|\mathit{astronaut}) = 0.5
\end{aligned}$$

Here we have:

$$\frac{1}{2}L_1(p,q) = 1 - \sum_i \min(p_i, q_i) = 1 - 0.5 = 0.5$$

So if we looked at the sets of adjectives that occurred with a large number of uses of *cosmonaut* and *astronaut* in a corpus, then the overlap of the two sets would be expected to be 0.5, corresponding to the proportion of occurrences of *spacewalking* with each noun.

Dagan et al. (1997b) compared the three dissimilarity measures (KL, IRad, and L_1) on a task similar to the selectional preferences problem in section 8.4. Instead of looking at the fit of nouns as argument of verbs, they looked at the fit of verbs as predicates for nouns. For example, given a choice of the verbs *make* and *take* the similarity measures were used to determine that *make* is the right verb to use with *plans* (*make plans*) and *take* is the right verb to use with *actions* (*take actions*).

[9] The following derivation shows that $\frac{1}{2}L_1(p,q) = 1 - \sum_i \min(p_i, q_i)$:

$$\begin{aligned}
L_1(p,q) &= \sum_i |p_i - q_i| \\
&= \sum_i \Big[\max(p_i, q_i) - \min(p_i, q_i)\Big] \\
&= \sum_i \Big[(p_i + q_i - \min(p_i, q_i)) - \min(p_i, q_i)\Big] \\
&= \sum_i p_i + \sum_i q_i - 2\sum_i \min(p_i, q_i) \\
&= 2\Big(1 - \sum_i \min(p_i, q_i)\Big)
\end{aligned}$$

Note that this also shows that $0 \leq L_1(p,q) \leq 2$ since $\sum_i \min(p,q) \geq 0$.

Here is how the similarity measure is used to compute the conditional probability $P(\text{verb}|\text{noun})$, which Dagan et al. (1997b) use as a measure of 'goodness of fit:'

$$P_{\text{SIM}}(v|n) = \sum_{n' \in S(n)} \frac{W(n, n')}{N(n)} P(v|n') \qquad (8.45)$$

Here, v is the verb, n is the noun, $S(n)$ is the set of nouns closest to n according to the similarity measure,[10] $W(n, n')$ is a similarity measure derived from the dissimilarity measure and $N(n)$ is a normalizing factor: $N(n) = \sum_{n'} W(n, n')$.

This formulation makes it necessary to transform the dissimilarity measure (KL, IRad or L_1) into the similarity measure W. The following three transformations were used.

$$W_{\text{KL}}(p, q) = 10^{-\beta D(p\|q)} \qquad (8.46)$$

$$W_{\text{IRad}}(p, q) = 10^{-\beta \,\text{IRad}(p\|q)} \qquad (8.47)$$

$$W_{L_1}(p, q) = (2 - L_1(p, q))^{\beta} \qquad (8.48)$$

The parameter β can be tuned for optimal performance.

Dagan et al. (1997b) show that IRad consistently performs better than KL and L_1. Consequently, they recommend IRad as the measure that is best to use in general.

This concludes our brief survey of measures of semantic similarity and dissimilarity. Vector space measures have the advantage of conceptual simplicity and of producing a similarity value that can be directly used for generalization. But they lack a clear interpretation of the computed measure. Probabilistic dissimilarity measures are on a more solid footing theoretically, but require an additional transformation to get to a measure of similarity that can be used for nearest neighbor generalization. Either approach is valuable in acquiring semantic properties of words from corpora by using similarity to transfer knowledge from known words to those that are not covered in the lexicon.

Exercise 8.19 [⋆]

Similarity-based generalization depends on the premise that similar things behave similarly. This premise is unobjectionable if the two uses of the word *similar* here refer to the same notion. But it is easy to fall into the trap of interpreting

[10] For the experiments, $S(n)$ was chosen to be the entire set of nouns, but one can limit the words considered to those closest to the target word.

them differently. In that case, similarity-based generalization can give inaccurate results.

Find examples of such potentially dangerous cases, that is, examples where words that are similar with respect to one aspect behave very differently with respect to another aspect.

Exercise 8.20 [⋆]

Similarity-based and class-based generalization are more closely related than it may seem at first glance. Similarity-based generalization looks at the closest neighbors and weights the input from these neighbors according to their similarity. Class-based generalization looks at the most promising class and, in the simplest case, generalizes the novel word to the average of that class. But class-based generalization can be made to look like similarity-based generalization by integrating evidence from all classes and weighting it according to how well the element fits into each class. Similarity-based generalization looks like class-based generalization if we view each element as a class.

Discuss the relationship between the two types of generalization. What role do efficiency considerations play?

Exercise 8.21 [⋆]

Co-occurrence matrices like the one in figure 8.3 represent different types of information depending on how co-occurrence is defined. What types of words would you expect *fire* to be similar to for the following definitions of co-occurrence: co-occurrence within a document; co-occurrence within a sentence; co-occurrence with words at a maximum distance of three words to the right; co-occurrence with the word immediately adjacent to the right. (See Finch and Chater (1994) and Schütze (1995) for two studies that show how the latter type of immediate co-occurrence can be used to discover syntactic categories.)

Exercise 8.22 [⋆⋆]

The measures we have looked at compare simple objects like vectors and probability distributions. There have also been attempts to measure semantic similarity between more complex objects like trees (see (Sheridan and Smeaton 1992) for one example). How could one measure the (semantic?) similarity between trees? How might such an approach lead to a better measure of semantic similarity between words than 'flat' structures?

Exercise 8.23 [⋆]

Select two words heading columns in figure 8.3 and compute pairwise similarities using each of the measures in table 8.7 for each of the three matrices in figures 8.3 through 8.5.

Exercise 8.24 [⋆]

Show that dice and cosine coefficients are identical if the two vectors compared have the same number of non-zero entries.

Exercise 8.25 [⋆]

Semantic similarity can be context-dependent. For example, electrons and tennis balls are similar when we are talking about their form (both have a round shape) and dissimilar when we are talking about their sizes.

Discuss to what extent similarity is context-dependent and when this can hinder correct generalization.

Exercise 8.26 [⋆]

Show that divergence to the average (IRad) is bounded by $2 \log 2$.

Exercise 8.27 [⋆]

Select two words heading columns in figure 8.3 and compute the three measures of dissimilarity in table 8.9 for each of the matrices in figures 8.3 through 8.5. You will have to smooth the probabilities for KL divergence. Are the dissimilarity measures asymmetric for KL divergence?

Exercise 8.28 [⋆⋆]

Both the L_1 norm and the Euclidean norm are special cases of the Minkowski norm L_p:

$$L_p(a,b) = \sqrt[p]{\sum_i |a_i - b_i|^p} \tag{8.49}$$

In this context, the Euclidean norm is also referred to as L_2. So the L_1 norm can be seen as a more appropriate version of the Euclidean norm for probability distributions.

Another norm that has been used for vectors is L_∞, that is L_p for $p \to \infty$ (Salton et al. 1983). What well-known function does L_∞ correspond to?

Exercise 8.29 [⋆]

Does a dissimilarity measure of 0 on one of the measures in table 8.9 imply that the other two measures are 0 too?

Exercise 8.30 [⋆]

If two probability distributions are maximally dissimilar according to one measure in table 8.9 (e.g., $\text{IRad}(p,q) = 2 \log 2$), does that imply that they are maximally dissimilar according to the other two?

8.6 The Role of Lexical Acquisition in Statistical NLP

Lexical acquisition plays a key role in Statistical NLP because available lexical resources are always lacking in some way. There are several reasons for this.

One reason is the cost of building lexical resources manually. For many types of lexical information, professional lexicographers will collect more accurate and comprehensive data than automatic procedures. But often manually constructed dictionaries are not available due to the cost of their construction. One estimate for the average time it takes to create a lexical entry from scratch is half an hour (Neff et al. 1993; obviously it depends on the complexity of the entry), so manual resource construction can be quite expensive.

There is one type of data that humans, including lexicographers, are notoriously bad at collecting: quantitative information. So the quantitative part of lexical acquisition almost always has to be done automatically, even if excellent manually constructed lexical resources are available for qualitative properties.

More generally, many lexical resources were designed for human consumption. The flip side of quantitative information being missing (which may be less important for people) is that the computer has no access to contextual information that is necessary to interpret lexical entries in conventional dictionaries. This is expressed aptly by Mercer (1993): "one cannot learn a new language by reading a bilingual dictionary." An example is the irregular plural *postmen* which is not listed as an exception in the lexical entry of *postman* in some dictionaries because it is obvious to a human reader that the plural of *postman* is formed in analogy to the plural of *man*. The best solution to problems like these is often the augmentation of a manual resource by automatic means.

Despite the importance of these other considerations motivating automated lexical acquisition, the main reason for its importance is the PRODUCTIVITY inherent productivity of language. Natural language is in a constant state of flux, adapting to the changing world by creating names and words to refer to new things, new people and new concepts. Lexical resources have to be updated to keep pace with these changes. Some word classes are more likely to have coverage gaps than others. Most documents will mention proper nouns that we have not encountered before whereas there will hardly ever be newly created auxiliaries or prepositions. But the creativity of language is not limited to names. New nouns and verbs also occur at a high rate in many texts. Words that are covered in the dictionary may still need the application of lexical acquisition methods because they develop new senses or new syntactic usage patterns.

How can we quantify the amount of lexical information that has to be learned automatically, even if lexical resources are available? For a rough

Type of coverage problem	Example
proper noun	*Caramello*, *Château-Chalon*
foreign word	*perestroika*
code	*R101*
mathematical object	x_1
non-standard English	*havin'*
abbreviation	*NLP*
hyphenated word	*non-examination*
hyphen omitted	*bedclothes*
negated adjective	*unassailable*
adverbs	*ritualistically*
technical vocabulary	*normoglycaemia*
plural of mass noun	*estimations*
other cases	*deglutition*, *don'ts*, *affinitizes* (VBZ)

Table 8.10 Types of words occurring in the LOB corpus that were not covered by the OALD dictionary.

assessment, we can consult Zipf's law and other attempts to estimate the proportion of as yet unseen words and uses in text (see chapter 6 and, for example, (Baayen and Sproat 1996) and (Youmans 1991)).

LEXICAL COVERAGE

A more detailed analysis is provided in (Sampson 1989). Sampson tested the coverage of a dictionary with close to 70,000 entries (the OALD, Hornby 1974) for a 45,000 word subpart of the LOB corpus. (Numbers were not counted as words.) He found that about 3% of tokens were not listed in the dictionary. It is instructive to look at the different types of words that are the cause of coverage problems. Table 8.10 lists the major types found by Sampson and some examples.

More than half of the missing words were proper nouns. The other half is due to the other categories in the table. Some of the coverage problems would be expected not to occur in a larger dictionary (some frequent proper nouns and words like *unassailable*). But based on Sampson's findings, one would expect between one and two percent of tokens in a corpus to be missing from even a much larger dictionary. It is also important to note that this type of study only gets at character strings that are entirely missing from the dictionary. It is much harder to estimate at what rate known words are used with new senses or in novel syntactic constructions. Finally, the one to two percent of unknown words tend to be among the most important in a document: the name of the person profiled in an article or the abbreviation for a new scientific phenomenon. So even if novel words constitute only a small percentage of the text, having an operational representation for their properties is paramount.

It took a long time until the limitations of dictionaries and hand-crafted knowledge bases for successful language processing became clear to NLP researchers. A common strategy in early NLP research was to focus on a small subdomain to attack what seemed to be the two most fundamental problems: parsing and knowledge representation. As a result of this focus on small subdomains, this early research "provided nothing for general use on large-scale texts" and "work in computational linguistics was largely inapplicable to anything but to sub-languages of very limited semantic and syntactic scope" (Ide and Walker 1992).

Problems of lexical coverage started to take center stage in the late eighties when interest shifted from subdomains to large corpora and robust systems, partly due to the influence of speech recognition research. One of the earliest pieces of work on lexical acquisition from corpora was done for the FORCE4 system developed by Walker and Amsler (1986) at SRI International. Since then, lexical acquisition has become one of the most active areas of Statistical NLP.

What does the future hold for lexical acquisition? One important trend is to look harder for sources of prior knowledge that can constrain the process of lexical acquisition. This is in contrast to earlier work that tried to start 'from scratch' and favored deriving everything from the corpus. Prior knowledge can be *discrete* as is the case when a lexical hierarchy like WordNet is used or *probabilistic*, for example, when a prior distribution over object noun classes is derived from a verb's dictionary entry and this prior distribution is then refined based on corpora. Much of the hard work of lexical acquisition will be in building interfaces that admit easy specification of prior knowledge and easy correction of mistakes made in automatic learning.

One important source of prior knowledge should be linguistic theory, which has been surprisingly underutilized in Statistical NLP. In addition to the attempts we have discussed here to constrain the acquisition process using linguistic insights, we refer the reader to Pustejovsky et al. (1993), Boguraev and Pustejovsky (1995), and Boguraev (1993) for work that takes linguistic theory as the foundation of acquisition. The last two articles summarize the important work on computational lexicography done at Cambridge University (described in detail in (Boguraev and

Briscoe 1989)), which, although mostly non-statistical, contains important insights on how to combine theoretical linguistics and empirical acquisition from lexical resources.

Dictionaries are only one source of information that can be important in lexical acquisition in addition to text corpora. Other sources are encyclopedias, thesauri, gazeteers, collections of technical vocabulary and any other reference work or data base that is likely to contribute to a characterization of the syntactic and semantic properties of uncommon words and names.

The reader may have wondered why we have limited ourselves to textual sources. What about speech, images, video? Lexical acquisition has focused on text because words are less ambiguous descriptors of content than features that can be automatically extracted from audio and visual data. But we can hope that, as work on speech recognition and image understanding progresses, we will be able to ground the linguistic representation of words in the much richer context that non-textual media provide. It has been estimated that the average educated person reads on the order of one million words in a year, but hears ten times as many words spoken. If we succeed in emulating human acquisition of language by tapping into this rich source of information, then a breakthrough in the effectiveness of lexical acquisition can be expected.

8.7 Further Reading

There are several books and special issues of journals on lexical acquisition: (Zernik 1991a), (Ide and Walker 1992), (Church and Mercer 1993), and (Boguraev and Pustejovsky 1995). More recent work is covered in later issues of *Computational Linguistics*, *Natural Language Engineering*, and *Computers and the Humanities*. In what follows, we point the reader to some of the work on lexical acquisition we were not able to cover.

Other approaches to the resolution of attachment ambiguity include transformation-based learning (Brill and Resnik 1994) and loglinear models (Franz 1997). Collins and Brooks (1995) used a back-off model to address data sparseness issues. Attachment ambiguity in noun phrases also occurs in Romance languages. See (Bourigault 1993) for French and (Basili et al. 1997) for Italian.

An alternative to Resnik's information-theoretic approach to the acquisition of selectional preferences is work by Li and Abe (1995) that uses a

Minimum Description Length framework. In (Li and Abe 1996), this work is extended to take into account the dependency between two or more arguments of a verb. For example, *drive* can take *car* as a subject (*This car drives well*), but only if there is no object. This type of regularity can only be discovered we we look at all arguments of the verb simultaneously. See also (Velardi and Pazienza 1989) and (Webster and Marcus 1989) for early (non-probabilistic, but corpus-based) work on selectional preferences.

Once we have acquired information about the selectional preferences of a verb, we can exploit this knowledge to acquire subcategorization frames, the first problem we looked at in this chapter. Poznański and Sanfilippo (1995) and Aone and McKee (1995) take this approach. For example, a verb that takes an NP of type 'beneficiary' or 'recipient' is likely to subcategorize for a *to*-PP.

Apart from semantic similarity, the automatic enhancement of hierarchies has been another focus in the area of acquiring semantics. Hearst and Schütze (1995) and Hearst (1992) describe systems that insert new words into an existing semantic hierarchy and Coates-Stephens (1993) and Paik et al. (1995) do the same for proper nouns. Riloff and Shepherd (1997) and Roark and Charniak (1998) assign words to categories assuming a flat category structure (which can be regarded as a simplified semantic hierarchy).

Two other important types of semantic information that attempts have been made to acquire from corpora are antonyms (Justeson and Katz 1991) and metaphors (Martin 1991).

We suggested above that non-textual data are a worthwhile source of information to exploit. There are some research projects that investigate how lexical acquisition could take advantage of such data once the problem of how to automatically build a representation of the context of an utterance has been solved. Suppes et al. (1996) stress the importance of action-oriented matching between linguistic forms and their contextual meaning (as opposed to acquiring word meaning from passive perception). Siskind (1996) shows that even if the contextual representation is highly ambiguous (as one would expect in a realistic learning situation), lexical acquisition can proceed successfully.

As a last source of information for acquiring meaning, we mention work on exploiting morphology for this purpose. An example of a morphological regularity that implies a particular type of meaning is the progressive tense. In English, only non-stative verbs occur in the progressive tense. Oversimplifying somewhat, we can infer from the fact that we find *he is running* in a corpus, but not *he is knowing* that *know* is stative and *run* is non-stative. See (Dorr and Olsen 1997), (Light 1996) and (Viegas et al. 1996) for work along these lines. While none of these papers take a statistical approach, such morphological information could be a fertile ground for applying statistical methods.

We conclude these bibliographic remarks by pointing the reader to two important bodies of non-statistical work that warrant careful study by anybody interested in lexical acquisition. They are of great potential importance either because they suggest ways of combining statistical approaches with symbolic approaches (as in the regular-expression post-filtering of collocations in (Justeson and Katz 1995b)) or because the insights they offer can often be expressed in a statistical framework as well as in a non-statistical framework, making them a valuable source for future statistical work.

The first area is the work on building syntactic and semantic knowledge bases from machine-readable dictionaries described by Boguraev and Briscoe (1989) and Jensen et al. (1993). These two books are a good starting point for those who want to learn about the strengths and weaknesses of dictionaries for lexical acquisition. We have focused on corpus-based acquisition here because that has been the bias in Statistical NLP, but we believe that most future work will combine corpus-based and dictionary-based acquisition.

The second area is the application of regular expression matching to natural language processing. (See (Appelt et al. 1993), (Jacquemin 1994), (Voutilainen 1995), (Sproat et al. 1996), and (Jacquemin et al. 1997) for examples.) There are phenomena and processing steps in lexical acquisition that deal with purely symbolic information and that can be well modeled in terms of regular languages. (Tokenization of English is an example.) In such cases, the speed and simplicity of finite state automata cannot be matched by other methods (Roche and Schabes 1997; Levine et al. 1992).