Proc. Natl. Acad. Sci. USA
Vol. 92, pp. 10052–10059, October 1995
Colloquium Paper

This paper was presented at a colloquium entitled "Human–Machine Communication by Voice," organized by Lawrence R. Rabiner, held by the National Academy of Sciences at The Arnold and Mabel Beckman Center in Irvine, CA, February 8–9, 1993.

New trends in natural language processing: Statistical natural language processing

MITCHELL MARCUS

Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104-6389

ABSTRACT **The field of natural language processing (NLP) has seen a dramatic shift in both research direction and methodology in the past several years. In the past, most work in computational linguistics tended to focus on purely symbolic methods. Recently, more and more work is shifting toward hybrid methods that combine new empirical corpus-based methods, including the use of probabilistic and information-theoretic techniques, with traditional symbolic methods. This work is made possible by the recent availability of linguistic databases that add rich linguistic annotation to corpora of natural language text. Already, these methods have led to a dramatic improvement in the performance of a variety of NLP systems with similar improvement likely in the coming years. This paper focuses on these trends, surveying in particular three areas of recent progress: part-of-speech tagging, stochastic parsing, and lexical semantics.**

SOME LIMITATIONS OF RULE-BASED NLP

Until about 3 or 4 years ago, all natural language processing (NLP) systems were entirely hand constructed, with grammars and semantic components made up of many carefully hand-crafted rules. Often the target coverage of such systems was based on a small set of exemplar sentences; many such systems were originally developed on fewer than several hundred examples. While these systems were able to provide adequate performance in interactive tasks with typed input, their success was heavily dependent on the almost magical ability of users to quickly adapt to the limitations of the interface.

The situation is quite different, however, when these rule sets are applied open loop to naturally occurring language sources such as newspaper texts, maintenance manuals, or even transcribed naturally occurring speech. It now appears unlikely that hand-coded linguistic grammars capable of accurately parsing unconstrained texts can be developed in the near term. In an informal study conducted during 1990 (1), short sentences of 13 words or less taken from the Associated Press (AP) newswire were submitted to a range of the very best parsers in the United States, parsers expressly developed to handle text from natural sources. None of these parsers did very well; the majority failed on more than 60% of the test sentences, where the task was to find the one correct parse for each sentence in the test set. Another well-known system was tested by its developer using the same materials in 1992, with a failure rate of 70%.

This failure rate actually conflates two different, and almost contradictory, problems of this generation of parsers. The first is that the very large handcrafted grammars used by parsers that aim at broad coverage often generate very large numbers of possible parses for a given input sentence. These parsers usually fail to incorporate some source of knowledge that will accurately rank the syntactic and semantic plausibility of parses that are syntactically possible, particularly if the parser is intended to be domain independent. The second problem, somewhat paradoxically, is that these parsers often fail to actually provide the correct analysis of a given sentence; the grammar of a natural language like English appears to be quite vast and quite complex.

Why can't traditional approaches to building large software systems, using techniques like divide and conquer, solve this last problem? The problem is not that the grammar developers are not competent or that there is a lack of effort; a number of superb computational linguists have spent years trying to write grammars with broad enough coverage to parse unconstrained text. One hypothesis is that the development of a large grammar for a natural language leads into a complexity barrier similar to that faced in the development of very large software systems. While the human grammatical system appears to be largely modular, the interaction of the subcomponents is still sufficient to cause the entire system to be unmanageably complex. The net result is that the grammatical system does not appear to decompose easily into units that a team can develop and then join together. In support of this view is the fact that almost all of the large grammars extant are the result of a single grammar developer working over a long period of time. If this conclusion is correct, an approach to developing NLP systems must be found other than careful handcrafting.

STATISTICAL TECHNIQUES: FIRST APPEARANCE

One of the first demonstrations that stochastic modeling techniques, well known in the speech-processing community, might provide a way to cut through this impasse in NLP was the effective application of a simple letter trigram model to the problem of determining the national origin of proper names for use in text-to-speech systems (2). Determining the etymology of names is crucial in this application because the pronunciation of identical letter strings differs greatly from language to language; the string *GH*, for example, is pronounced as a hard *G* in Italian, as in *Aldrighetti*, while most often pronounced as *F* or simply silent in English, as in *laugh* or *sigh*. This system estimates the probability that a name W comes from language L as the product of the probabilities, estimated across a set of known names from L, of all the contiguous three-letter sequences in W. It then assigns W to the language L, which maximizes this probability. The success of this program came as a surprise to most of the NLP community, at the time completely wedded to the symbolic techniques of traditional artificial intelligence (AI). Many people in NLP thought this application was a fluke, that the task solved by the program was somehow special. In fact, this technique led the

The publication costs of this article were defrayed in part by page charge payment. This article must therefore be hereby marked "*advertisement*" in accordance with 18 U.S.C. §1734 solely to indicate this fact.

way toward application of statistical techniques to problems that one would have thought required an "AI-complete" solution, a full solution to the problem of modeling human understanding.

In another sense this work was an application of an approach to linguistic analysis called distributional analysis (3), which reached its zenith in the 1950s. This work suggested that the structure of language could be discovered by looking at distributional patterns of linguistic entities. While the work of Chomsky in the late 1950s showed that distributional analysis could not be the whole story, most linguists assumed that Chomsky's work implied that distributional techniques should be abandoned entirely. This application showed that simple distributional techniques were useful for solving hard engineering problems that looked resistant to the application of *a priori* knowledge.

THE ARCHITECTURE OF AN NLU SYSTEM

Fig. 1*a* gives an overview of a few of the crucial steps in the process of decoding a sentence in a conventional NLU system, given that the words that make up the sentence have been determined either by a speech recognition system or by tokenization of an ASCII source. When a new sentence comes in, it is analyzed by a parser that both determines what part of speech to assign to each of the words and combines these part-of-speech tagged words into larger and larger grammatical fragments, using some kind of grammar that tells what combinations are possible and/or likely. The output of this grammatical analysis, either a single-rooted tree or a string of tree fragments, then goes through semantic analysis, which determines the literal meaning of a sentence in isolation. This phase of analysis decides both what the individual words mean and how to combine the individual word meanings into larger

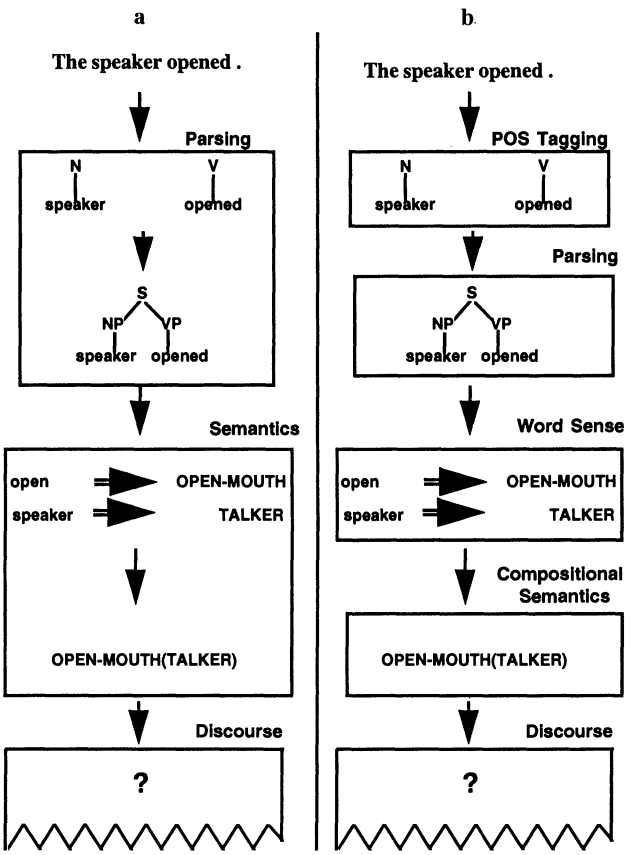

FIG. 1. Two decompositions of the architecture of an NLU system. (*a*) Standard decomposition. (*b*) An alternate decomposition.

semantical structures. Often, this last step is done using some form of *compositional semantics*, where the meaning of each larger unit is constrained to be a relatively simple function of the meaning of its parts. This meaning representation is then further analyzed by pragmatic and discourse components to determine what the sentence means given its particular context of use and to place this representation into a multisentence representation useful for such tasks as determining the referent of pronouns and other noun phrases.

For the purposes of the rest of this article, the problem will be subdivided into somewhat smaller functional units than given by the conventional model. This subdivision, given in Fig. 1*b*, reflects the development of statistical NLP techniques over the past several years, with rate of progress roughly proportional to height in the figure. The first success of statistical modeling techniques for NLU was in the area of part-of-speech determination—deciding, for example, whether the word *saw* functioned in a given linguistic context as a singular noun or a past tense verb. A variety of techniques now tag previously unseen material with 95 to 97% accuracy. Recently, purely context-free probabilistic parsing methods have been supplanted by parsing algorithms that utilize probabilities of context-free rules conditionalized on aspects of surrounding linguistic structure. Such parsers provide the correct parse as the first parse output between 60 to 80% of the time, by sentence, on naturally occurring texts from rich, but not entirely unconstrained, domains such as the *Wall Street Journal*. They have performed with up to 91% accuracy on spoken language tasks from limited domains like the Advanced Research Projects Agency's (ARPA) Air Travel Information Service (ATIS) domain. In the area of lexical semantics, a range of promising techniques for performing word-sense disambiguation have emerged recently, as well as some preliminary work in automatically determining the selectional restrictions of verbs, that is, what kind of objects can serve as the subject or object of a given verb.

Finally, all of these methods depend crucially on the availability of training materials annotated with the appropriate linguistic structure. These advances were made possible by the development of corpora appropriately annotated with part-of-speech and syntactic structure. This paper will also touch on the development of such corpora.

PART-OF-SPEECH TAGGING

The task of a part-of-speech tagger is to map an input stream of word tokens into the correct part of speech for each word token in context. To do this, it must disambiguate words that have the potential to be many different parts of speech. A part-of-speech tagger, for example, should map the sentence *Can we can the can*? into the string of parts of speech shown in Fig. 2. This problem of lexical disambiguation is a central problem in building any NLP system; given a realistically large lexicon of English, many common words are used in multiple parts of speech. Determining what function each word plays in context is a crucial part of either assigning correct grammatical structure for purposes of later semantic analysis or of providing a partial heuristic chunking of the input into phrases for purposes of assigning intonation in a text-to-speech synthesizer.

The problem of lexical disambiguation was thought to be completely intractable 10 years ago by most of the NLP community, and yet now a wide range of very different techniques can solve this problem with 95 to 97.5% word accuracy, for part-of-speech tag sets of between 40 and 80 tags,

Words in:	Can we can the can?
Part-of-speech stream out:	*modal pronoun verb det noun*

FIG. 2. Part-of-speech taggers assign tags in context.

depending on the task and how accuracy is measured (see, e.g., refs. 4–10 and others). It is worth noting that many of these errors are not very harmful; a significant fraction of the errors consist of cases where one kind of verb is confused with another kind of verb or one kind of noun with another. Many of the parsers for which these taggers serve as preprocessors are forgiving enough that the errors do not actually throw the parser off track.

Most part-of-speech taggers are implemented as hidden Markov models (HMMs). For an input sentence $S = w_1, w_2, \ldots w_n$, these taggers predict a tag t_i for each word w_i given two sets of probabilities: First, $P(w|t)$ (the probability of w given t), the probability for each possible word w and each part-of-speech tag t that if a given word is tagged with t, it is in fact the word w. Second, $P(t_{i+1})|t_i)$, the transition probability that the next tag is t_{i+1}, given that the current tag is t_i. These taggers use a linear time search utilizing the dynamic programming algorithm, often called the Viterbi algorithm, to find the string of tags $T = t_1, t_2, \ldots t_n$ that maximize $\Pi_i P(w_i|t_i) P(t_i|t_{i-1})$.

The question here is how to estimate the value of the parameters of the HMM. The standard approach for HMMs is to use the forward/backward algorithm to automatically estimate the parameters, as described by Jelinek (11) elsewhere in this volume. However, systems that use the forward/backward algorithm do not perform quite as well as those that estimate parameters, at least initially, by simple counting, using a corpus of text that has been pretagged with part-of-speech information (9). In practice, such systems must use some technique to *smooth*[*] very small counts. One could then use the forward/backward algorithm to smooth these direct estimates, but there is little evidence that this helps. Currently, then, the best way to estimate the parameters of an HMM for part-of-speech tagging is to hand tag a corpus and simply count.

The theme that emerges here is true of most statistical NLP applications and will be a leitmotif in what follows below. What works best both for part-of-speech tagging using HMMs and for the entire range of statistical NLP applications considered in this paper, is some appropriate combination of stochastic techniques and linguistic knowledge. While earlier work provides evidence that handcrafted symbolic representations of linguistic knowledge are insufficient to provide industrial-strength NLP, it also appears that the use of statistical methods without some incorporation of linguistic knowledge is insufficient as well. This linguistic knowledge may either be represented in *implicit* form, as in the use of a pretagged corpus here, or encoded *explicitly* in the form of a grammar.[†] In the next few years, I believe we are going to see stochastic techniques and linguistic knowledge more and more deeply interleaved.

The Problem of Unknown Words

In conjunction with this observation it is important to realize that if one simply implemented an HMM for part-of-speech tagging as discussed above, the performance of the resulting system on new material could well be no better than 70 or 80% correct. Without exception, input is preprocessed before parts of speech are assigned by an HMM; this preprocessing is often only partially discussed in technical descriptions of part-of-speech taggers. The preprocessing copes with "unseen words," words that were never seen in the training data and for which the system therefore has no prior knowledge of possible parts of speech. It turns out that about half of the word types in the Brown corpus (12, 13), a carefully balanced representative corpus of American English, appear exactly once (about 32,000 out of 67,000 word types). This is consistent with Zipf's law, the empirical law that the frequency of a word type is inversely proportional to its rank. Nor can this problem be circumvented by some appropriately huge lexicon; a very large number of proper names appear on any newswire for the first time each day.

How can this unseen word problem be handled? One simple but quite effective technique is to tag each unknown word with the most likely tag given its last three letters—an empirical approximation to simple morphological analysis (5). A useful heuristic for proper nouns in most English text is to use capitalization, often combined with some other heuristics to correct for unknown words used at the beginning of sentences (10). The key point here is that these techniques for unseen words go beyond using purely stochastic techniques to using implicit and explicit linguistic knowledge, although in a trivial way, to get the job done.

STOCHASTIC PARSING

All work on stochastic parsing begins with the development of the inside/outside algorithm (14), which generalizes the Baum-Welch algorithm for estimating HMMs to the estimation of parameters of stochastic context-free grammars.[‡] Just as each iteration of the Baum-Welch algorithm over some training corpus improves estimates of the parameters of the underlying HMM, as judged by the criterion of maximal likelihood, so the inside/outside algorithm improves parameter estimates of an underlying probabilistic context-free grammar, judged by this same criterion.

However, straightforward application of the inside/outside algorithm does not appear to produce effective parsers; the best results to date have resulted in parsers with about 35% correct parses on fully reserved test material in simple parsing tasks (17, 18). Two problems appear to lie behind this failure. First, for realistic probabilistic context-free grammars (PCFGs) the number of parameters that must be estimated is very large; unless some *a priori* constraint is provided, n^3 parameters must be estimated for a grammar with n nonterminal categories, categories that label not words but structures, like *noun phrase*, *verb phrase*, or *sentence*.

But a worse problem is that the objective function that the inside/outside procedure maximizes, namely the probability of the training corpus given the grammar, is in fact not the objective function that one wants to maximize to train effective parsers. For parsing the goal is to maximize assignment of the *correct grammatical structure*, to recursively subdivide the sentence correctly into its constituent grammatical parts, determined, say, by examining similar sentences in a treebank of hand-parsed sentences. Unfortunately, there is no reason to expect that a PCFG whose parameters are estimated by the inside/outside algorithm will assign structures that have the desired constituent structure.

In recent years a range of new grammatical formalisms have been proposed that some suggest have the potential to solve a major part of this problem. These formalisms, called *lexicalized* grammar formalisms, express grammars in which the entire grammar consists of complex structures associated with individual words, plus some very simple general rules for com-

[*]Since sentence probabilities are estimated by multiplying together many estimates of local probabilities, any probability estimate of zero leads to a zero probability for the entire string. Since any direct estimate is based on only finite data, it is important to assume that any event not observed at all has some very small, but nonzero probability. How to best perfrom this smoothing of probability estimates is a central technical issue in applying any of the methods discussed in this chapter.

[†]For readers familiar with logic, this is the distinction between knowledge represented *extensionally* and knowledge represented *intensionally*.

[‡]For a tutorial introduction to probabilistic context-free grammars and the inside/outside algorithm, see refs. 15 and 16.

bining these structures. Such grammar formalisms include combinatory categorial grammars (CCGs), lexicalized tree-adjoining grammars (LTAGs), and link grammars (19, 20). In these lexicalized formalisms each word can be thought of as a tree fragment; the full grammatical analysis of a sentence is formed by specifying how and in what order the fragments associated with each word in a sentence combine. Words may themselves be ambiguous between different "parts of speech," here differing tree fragments. In these grammar formalisms the bulk of parsing a sentence is just deciding on which part of speech to assign to each word. Given this property of these grammar formalisms, perhaps some way can be found to extend the inside/outside algorithm appropriately so that its objective function maximizes the probabilities of strings of *part-of-speech tagged* words. If so, it is just a matter of extending the search space to handle the large number of complex part-of-speech structures of lexicalized grammars.§

Constraining the Inside/Outside Algorithm

Recently, a number of experiments have been performed that combine the inside/outside algorithm with some form of linguistic knowledge. In a recent experiment by Pereira and Schabes (21), a modified version of the inside/outside algorithm was applied to a corpus that was manually annotated with a skeletal syntactic bracketing by the Penn Treebank Project (22, 23). In this experiment the I/O algorithm was modified to consider only PCFG rules that did not violate the skeletal bracketing of the corpus, zeroing out many of the n^3 parameters a priori. The algorithm was then trained on a corpus of only 770 sentences collected in the Air Travel Information System (ATIS) domain (24). The evaluation was based on the "crossing brackets" parser evaluation metric of Black *et al.* (25). This crossing-brackets measure counts the number of brackets inserted during parsing that are consistent with the correct bracketing of the sentence.¶ Without constraint, the algorithm achieved 35% bracketing accuracy on reserved test materials but achieved 90% bracketing accuracy when constrained by the annotated corpus.

Conditioning PCFG Rules on Linguistic Context

One new class of models uses linguistic knowledge to condition the probabilities of standard probabilistic context-free grammars. These new models, which in essence augment PCFG grammar rules with probabilistic applicability constraints, are based on the hypothesis that the inability of PCFGs to parse with high accuracy is due to the failure of PCFGs to model crucial aspects of linguistic structure relevant to the appropriate selection of the next grammar rule at each point within a context-free derivation. Probabilities in the standard stochastic context-free model are conditioned only on the type of nonterminal that the grammar is about to expand; the key idea of these new models is that this provides insufficient linguistic context to adequately model the probabilities of rule expansion. One such parser, that of Magerman and Marcus (26, 27), assumes that expansion of any nonterminal is conditioned on the type of nonterminal, the most likely part-of-speech assignments for the next several words in the parser's input stream, and the rule that has generated the particular nonterminal that the parser is trying to expand. For example, the rule "NP → pronoun" might have a different probability when it expands the NP in the rule "S → NP VP" than when it expands the NP in the rule "VP → NP NP"). Tested on a corpus of sentences from the Massachusetts Institute of Technology's Voyager domain (28), this parser correctly parsed 89% of a reserved test set. A sample list of sentences from this corpus, with length distribution typical of the corpus as a whole, is given in Fig. 3. Although the performance of this algorithm is quite impressive in isolation, the sentences in this corpus are somewhat simpler in structure than those in other spoken language domains and are certainly much simpler than sentences from newswire services that were the target of the parser evaluation discussed in the introduction to this article.

On the other hand, a simple PCFG for this corpus parses a reserved test set with only about 35% accuracy, comparable to PCFG performance in other domains. If the probability of each rule is conditioned on both the current nonterminal and on the particular rule that gave rise to the current nonterminal, then performance improves to about 50% accuracy. Conditioning each rule on the expected part of speech of the next several words in addition increases performance to 87.5% accuracy. The key point here again is that combining a very simple stochastic framework with a little bit of linguistic knowledge greatly increases performance over each alone.

Many parsing techniques are now emerging that combine stochastic techniques with linguistic knowledge in a number of different ways. Again, as discussed briefly above, linguistic knowledge can be encoded *explicitly*, perhaps in the form of a grammar, or *implicitly* within the annotations of an annotated corpus.

In combination with stochastic methods, so-called covering grammars can be used, grammars that provide at least one correct parse for sentences of interest but that may also produce spurious impossible parses. While these spurious parses would be a problem if the grammar were used with a purely symbolic parser, the hope is that when used within a stochastic framework, spurious parses will be of much lower probability than the desired analyses. One simple method for combining explicit linguistic knowledge with stochastic techniques is to use a stochastic technique to estimate the probability distribution for all and only the rules within the grammar, drastically limiting the number of parameters that need to be estimated within the stochastic model. While advocated by many researchers, this method suffers from the potential defect that it cannot model grammar rules that the grammar writer overlooked or that occur rarely enough that they were unseen in training materials. A somewhat more powerful method is to (*i*) use the grammar to generate all potential parses of a set of example sentences, (*ii*) create a training set of trees by either hand picking the correct parse for each sentence or simply using all potential parses (which works far better than might be expected), and then (*iii*) use the usage count of each grammar rule within this training set to provide an initial estimate of the parameters of the associated stochastic grammar, which might then be smoothed using the inside/outside algorithm.

I m currently at MIT
Forty-five Pearl Street
What kind of food does LaGroceria serve
Is there a Baybank in Central Square
Where is the closest library to MIT
What s the address of the Baybank near Hong Kong
What s the closest ice cream parlor to Harvard University
How far is Bel Canto s from Cambridge Street in Cambridge
Is there a subway stop by the Mount Auburn Hospital
Can I have the phone number of the Cambridge City Hall
Can you show me the intersection of Cambridge Street and Hampshire Street
How do I get to the closest post office from Harvard University
Which subway stop is closest to the library at forty-five Pearl Street

FIG. 3. Sample sentences from the Massachusetts Institute of Technology's Voyager corpus.

§I thank Aravind Joshi for the above observation.
¶Notice that this is a much easier measure than the percentage of sentences parsed correctly; if one of, say, 33 brackets is inconsistent in a given sentence, the sentence is 97% correct by the bracket measure and simply wrong by the sentences-correct measure.

Associated with food (y = food; fy = 2240)				Associated with water (y = water; fy = 3574)			
I(x;y)	fxy	fx	x	I(x;y)	fxy	fx	x
9.62	6	84	hoard	9.05	16	208	conserve
8.83	9	218	go_without	8.98	18	246	boil
7.68	58	3114	eat	8.64	6	104	ration
6.93	8	722	consume	8.45	10	198	pollute
6.42	6	772	run_of	8.40	20	408	contaminate
6.29	14	1972	donate	8.37	38	794	pump
6.08	17	2776	distribute	7.86	6	178	walk_on
5.14	51	15900	buy	7.81	43	1320	drink
4.80	53	21024	provide	7.39	15	618	spray
4.65	13	5690	deliver	7.39	9	370	poison

Computed over Parsed AP Corpus (N = 24.7 million SVO triples)

FIG. 4. What do you typically do with *food* and *water* (32)?

If the annotation of a corpus takes the form of syntactic bracketing, the implicit knowledge encoded in the annotations of a corpus can be used in exactly the same way as explicit knowledge, to a first approximation. The key idea is that a grammar can be simply extracted from the corpus, along with counts for each rule, and then the methods discussed immediately above are applicable. In fact, most grammatical annotation provided in corpora to date is *skeletal*, providing only partial structures, so the use of smoothing techniques is crucial.

To what extent might combinations of linguistic knowledge with stochastic techniques improve the performance of parsers in the near-term future? Two experiments, both on a corpus of short sentences from computer manuals, cast some light here. In the first experiment (1), both an explicit grammar and an annotated corpus were used to build a stochastic parser that parsed 75% of the sentences in a reserved test set completely consistently with a hand-assigned bracketing. The second experiment (29) is attempting to leave explicit grammars behind, using instead a very rich set of linguistically relevant questions in combination with decision tree techniques. These questions examine not only syntactic properties, but lexical and class-based information as well, thus combining a much richer set of linguistic knowledge sources than any other model to date. The decision tree uses this set of questions to search for the grammar implicit in a very large hand-annotated corpus. Published reports of early stages of this work indicate that this technique is 70% correct on computer manual sentences of length 7 to 17, where, to count as correct, each parse must exactly match the prior hand analysis of that sentence in the test corpus, a more stringent test criterion than any other result mentioned here.

While this last experiment uses one uniform statistical technique, decision trees, to make all parsing decisions, some recent work suggests that effective parsing might be done by a suite of interacting parsing experts, each handling a particular grammatical phenomenon. Perhaps the clearest example of this is a recent technique to resolve the ambiguous attachment of prepositional phrases. Consider the sentence *I saw the man with the telescope*; here the prepositional phrase *with the telescope* might modify *the man*, meaning *I saw the man who had a telescope*, or it might modify the main verb *saw*, meaning *I used the telescope to see the man*. If the sentence were instead *I saw the planet with the telescope*, the prepositional phrase would certainly modify the main verb, but if it were *I saw the man with the hat* the prepositional phrase would clearly modify *the man*. Here, as in many other cases, it becomes clear that a decision about grammatical structure depends crucially on the properties of the lexical items themselves. A technique that uses likelihood ratios to compare the strength of association between the preposition and the main verb with the strength of association between the preposition and the preceding noun correctly assigns about 80% of prepositional phrases in sentences from the AP newswire with structure identical to the examples here (30). It is interesting to note that human judges, given the same information, do this task at about 85 to 87% accuracy. This experiment also points out the key role of lexical properties in deciding grammatical structure. Its success suggests that the crucial role of grammar is just to mediate the properties of lexical items themselves. This would suggest, as

More with food				More with water			
t	food	water	w	t	food	water	w
7.47	58	1	eat	−6.93	0	50	be_under
6.26	51	7	buy	−5.62	1	38	pump
4.61	31	6	include	−5.37	3	43	drink
4.47	53	25	provide	−5.20	0	29	enter
4.18	31	9	bring	−4.87	1	30	divert
3.98	21	3	receive	−4.80	0	25	pour
3.69	14	0	donate	-4.25	0	20	draw
3.55	13	0	prepare	−4.01	0	18	boil
3.31	13	1	offer	−3.89	0	17	fall_into
3.08	13	2	deliver	−3.75	1	20	contaminate

Computed over Parsed AP Corpus (N = 24.7 million SVO triples)

FIG. 5. What do you do more with *food* than with *water* (32)?

does the recent work on lexicalized grammars discussed above, that the words themselves are primary.

Annotated Corpora

Before leaving the question of syntax, I would like to say a word about the production of annotated corpora themselves. There are, at the moment, two large grammatically annotated corpora for English—the IBM/Lancaster Treebank (31) and the Penn Treebank (23). As of this time, only materials from the second are generally available; they are distributed through the Linguistic Data Consortium; because of this, and my familiarity with this corpus, that is the focus here.

The Penn Treebank now consists of 4.5 million words of text tagged for part of speech, with about two-thirds of this material also annotated with a skeletal syntactic bracketing. All of this material has been hand corrected after processing by automatic tools. The two largest components of the corpus consist of over 1.6 million words of material from the Dow-Jones News Service, hand parsed, with an additional 1 million words tagged for part of speech and a skeletally parsed version of the Brown corpus (12, 13), the classic 1-million-word balanced corpus of American English. This material has already been used for purposes ranging from serving as a gold standard for parser testing to serving as a basis for the induction of stochastic grammars to serving as a basis for quick lexicon induction.

There is now much interest in very large corpora that have quite detailed annotation, assuming that such corpora can be efficiently produced. The group at Penn is now working toward providing a 3-million-word bank of predicate-argument structures. This will be done by first producing a corpus annotated with an appropriately rich syntactic structure and then automatically extracting predicate-argument structure, at the level of distinguishing logical subjects and objects, and distinguishing a small range of particular adjunct classes. This corpus will be annotated by automatically transforming the current treebank into a level of structure close to the intended target and then completing the conversion by hand.

LEXICAL SEMANTICS AND BEYOND

We now turn to an area of very recent progress—lexical semantics. At initial inspection, it would appear most unlikely that statistical techniques would be of much use for either the discovery or representation of the meanings of words. Surprisingly, some preliminary work over the past several years indicates that many aspects of lexical semantics can be derived from existing resources using statistical techniques.

Several years ago it was discovered that methods from statistics and information theory could be used to "tease out" distinctions between words, as an aid to lexicographers developing new dictionaries (32). As an example, consider the following: How could one distinguish the meaning of *food* and *water*? Fig. 4 shows the mutual information score,‖ an information theoretic measure, between various verbs and between *food* and *water* in an automatically parsed corpus, where either *food* or *water* is the object of that verb or, more precisely, where one or the other is the head of the noun phrase which is the object of the verb. The corpus used in this experiment consists of 25 million subject-verb-object triples automatically extracted from the AP newswire by the use of a parser for unrestricted text. The mutual information score is high if the verb and noun tend to occur together and will tend toward 0 if the verb and noun occur together no more often than expected by chance. Because this measure is the log of a ratio, scores such as those shown in the table are quite high. What

‖The mutual information statistic is a measure of the interdependence of two signals. It is defined as $MI(x, y) = \log [P(x, y)/P(x)P(y)]$.

Word:	prendre
Informant:	Right noun
Information:	.381 bits

a For *prendre* the noun to the right is maximally informative.

Sense 1	Sense 2
part	décision
mesure	parole
note	connaissance
exemple	engagement
temps	fin
initiative	retraite

b Same French words that are informant values for each sense.

Pr(English/Sense 1)		Pr(English/Sense 2)	
to_take	.433	to_make	.186
to_make	.061	to_speak	.105
to_do	.051	to_rise	.066
to_be	.045	to_take	.066
		to_be	.058
		decision	.036
		to_get	.025
		to_have	.021

c Sense one translates as *take*, sense two as *make*.

FIG. 6. The two senses of *Prendre* translate as *take* or *make* (33).

kinds of things can you do with food? Well, according to the AP newswire, you can hoard it, go without it, eat it, consume it, etc. With water, you can conserve it, boil it, ration it, pollute it, etc.** This indeed begins to reveal something about the meaning of these verbs, based on *distributional* properties of these words, that is, what other words they cooccur with.

To differentiate these two words somewhat more sharply, one might ask what might be done more or less to one than the other. Here, an appropriate metric is the t-score, which contrasts the conditional probability of seeing food as object given a particular verb with the conditional probability of seeing water as object given that verb.†† Fig. 5 shows that one eats or buys food far more than water and that one pumps or drinks water far more than food. Perhaps surprisingly, these descriptions get quite close to the heart of the difference between *food* and *water*.

These experiments show that statistical techniques can be used to tease out aspects of lexical semantics in such a way that a human lexicographer could easily take advantage of this information. However, for computers to utilize such information, some kind of representation must be found to encode semantical information in the machine. What kinds of representations might be appropriate for automatic discovery procedures? Much research on automatic machine translation is now being done using the parallel French-English transcripts of the proceedings of the Canadian parliament. This corpus has been used to test a statistical technique that finds the most reliable local clue in context to tease apart different senses of the same word in the source language, representing that meaning as the translation in the target language (33). An example of this technique is shown in Fig. 6. Here, the most useful single clue for the translation of an instance of the French word *prendre* is the particular word that

**Because English contains a large number of so-called phrasal verbs, whenever the parser encounters a verb followed immediately by a prepositional phrase, as in *go without food*, the parser creates a potential phrasal verb, for example, *go_without* and a triple where the object of the preposition (here *food*) is taken as the object of the putative phrasal verb (here *go_without*).

††The formula computed is $t = [P(food|verb) - P(water|verb)]/\{s^2[P(food|verb)] + s^2[P(water|verb)]\}^{1/2}$.

occurs as the first noun to its right. As shown in Fig. 6*a*, the identity of this word provides, on average, 0.381 bits of information as to how to distinguish between two different senses of *prendre*. Fig. 6*b* shows some of the French words that distinguish between one sense of *prendre* (*part*, *mesure*, *note*, *exemple*, etc.) and another (*décision*, *parole*, *connaissance*, etc.). As shown in Fig. 6*c*, the most likely translation into English for sense 1 is the verb *take*; for sense 2, the most likely translation is the English verb *make*. This technique has shown itself to be quite useful in current work in machine translation. As a technique for word sense disambiguation, it has the clear drawback that often a word with multiple senses in language 1 has many of the same senses in language 2, so it can only be used when the target language *does* split the senses of interest.

Other researchers have found a richer and more robust representation for words in the internal representation used within WordNet, a large hand-built computer-based lexicon (34). Because WordNet is fundamentally computer based, it can be organized as a large graph of different kinds of relations between words. These relations include not only relatively standard ones such as *X is a synonym of Y*, or *X is an antonym of Y* but also many other relations such as *X is a kind of Y* and *X is a part of Y*. Concepts within WordNet are represented by "synonym sets," sets of words that all share a core meaning. One simple representation of a meaning of a word, then, is just the synonym set, or *synset*, of the words that share that meaning.

This hierarchy has been used to investigate the automatic classification of verbs by the kinds of objects that they take, a first step toward determining the *selectional restrictions* of verbs automatically (35). In this work, synonym sets are used to represent classes of objects in both the input and output of a program that computes a variant of the mutual information statistic discussed above. Using synonym sets for the output provides the general classification one seeks, of course. Using synonym sets for input as well has an important added advantage: it provides a solution to the sparse data problem that plagues work in lexical statistics. Many of the counts of verb-object pairs that make up the input to this program are very small and therefore unreliable, in particular given a corpus as small as the million-word Brown corpus (12, 13) used in this experiment. By pooling data for particular nouns into the synonym sets they fall into, much of this sparse data problem can be solved.

Fig. 7 gives one example of the performance of Resnik's statistic. These are the highest-ranking synonym sets for objects of the verb *open*. The names of the synonym sets were hand assigned within WordNet. Fig. 8 gives the single highest ranking synonym set for a list of common verbs. These two experiments show that a statistical approach can do surprisingly well in extracting major aspects of the meaning of verbs, given the hand encoding of noun meanings within WordNet. These experiments suggest that it might be possible to combine the explicit linguistic knowledge in large hand-built computational lexicons, the implicit knowledge in a skeletally parsed corpus, and some novel statistical and information theoretic methods to automatically determine a wide variety of aspects of lexical semantics.

Verb	Most Highly Associated Object SynSet
ask	question
cell	someone
climb	stair
cook	repast
draw	cord
drink	beverage
eat	nutrient
lose	sensory_faculty
play	part
pour	liquid
pull	cover
push	button
read	written_material
sing	music

FIG. 8. "Prototypical" classes of objects for common verbs (35).

The work described above is typical of much recent work in the area of lexical discovery. Other recent work has focused on, for example, the use of distributional techniques for discovery of collocations in text (36) and of subcategorization frames for verbs (37) and to uncover lexical semantics properties (38). Much other work has been done in this area; the references given here are typical rather than exhaustive.

A QUESTION FOR TOMORROW

While the focus above has been on the effectiveness of recent stochastic and statistical techniques, there is some evidence that this effectiveness is due in large measure to the *empirical corpus-based* nature of these techniques rather than to the power of stochastic modeling. Surprisingly, symbolic learning techniques have performed as well as stochastic methods on two tasks considered above, despite the fact that they learn only simple symbolic rules, with only simple counting used during training, and then only to choose one potential rule over another. This raises the question of whether the effectiveness of the stochastic techniques above is essentially due to the fact that they extract linguistic structure from a large collection of natural data or is the result of their statistical nature. This issue, I believe, will be resolved in the next several years.

In one experiment a very simple symbolic learner integrated with a parser for free text produced a set of symbolic lexical disambiguation rules for that parser. The parser, running with the new augmented grammar, if viewed only as a part-of-speech tagger, operates at about 95% word accuracy (8). What makes this result all the more surprising is that this parser works strictly left to right in a fully deterministic fashion. Recently, a very simple symbolic learning technique called error-based transformation learning was applied to the tagging problem (5). The resultant tagger operates with a set of only

SynSet Name	Typical members
entrance	door
mouth	mouth
repository	store, closet, locker, trunk
container	bag, trunk, locker, can, box, hamper
time_period	tour, round, season, spring, session, week, evening, morning, saturday
oral_communication	discourse, engagement, relation, reply, mouth, program, conference, session
writing	scene, book, program, statement, bible, paragraph, chapter

FIG. 7. Classes of things that are opened (35).

160 simple rules, plus a table of the most likely tag in isolation for each word. The tagger begins by tagging each word with the most likely tag in isolation and then applies each of the 160 rules in order to the entire corpus. These rules are of the form *In a context X, change tag A to tag B*. This tagger operates at about 96% word accuracy.

This learning algorithm has also been applied to the problem of bracketing English text, with very surprising results (39). The learner begins by assuming that English is strictly right branching and then learns a set of rules using exactly the same learning technique as used in the tagger discussed above, except that here the potential environments for rule application are very simple configurations in the bracketed string, for example, *if some category X is preceded by a right paren then . . . or if a left paren falls between category X and category Y then. . . . There are only two possible rule operations that simply transform the binary branched trees with bracketings* ((*A B*) *C*) and (*A* (*B C*)) into each other. The parser was trained on a small 500-sentence bracketed subset of material from the *Wall Street Journal*, of sentences less than 15 words in length, and acquired about 160 rules. Tested on a reserved test set of sentences of the same length, the parser bracketed 54% of the sentences completely consistently with the original bracketing; 72% of the sentences were bracketed with one bracketing error or less. Trained on only 250 sentences of length n, $n \leq 25$, the parser again acquired about 160 rules and parsed a similar reserved test set at 30% of sentences bracketed correctly. In recent unpublished experiments this same technique was applied to the problem of labeling these bracketed but unlabeled structures, achieving about 95% correct labeling, by node.

Perhaps this learning technique will lead to an even more powerful stochastic method of some kind. What is unique about this learner is that each rule applies to the output of the previous rule. But perhaps it will turn out that the power of these methods comes from use of a corpus itself. Time will tell.

This work was partially supported by Defense Advanced Research Projects Agency (DARPA) Grant No. N0014-85-K0018, by DARPA and (AFOSR) jointly under Grant No. AFOSR-90-0066, and by Grant No. DAAL 03-89-C0031 PRI. Thanks to Eric Brill, Ken Church, Aravind Joshi, Mark Liberman, David Magerman, Yves Schabes, and David Yarowsky for helpful discussions. Thanks also to two anonymous reviewers for many excellent comments and suggestions.

1. Black, E., Jelinek, F., Lafferty, J., Magerman, D. M., Mercer, R. & Roukos, S, in *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*.
2. Church, K. (1985) in *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, pp. 246–253.
3. Harris, Z. (1951) *Methods in Structural Linguistics* (Univ. of Chicago Press, Chicago).
4. Black, E., Jelinek, F., Lafferty, J., Mercer, R. & Roukos, S. (1992) in *Proceedings of the DARPA Speech and Natural Language Workshop*, February, pp. 117–121.
5. Brill, E. (1992) in *Proceedings of the Third Conference on Applied Natural Language Processing* (Trento, Italy).
6. Church, K. (1988) in *Proceedings of the Second Conference on Applied Natural Language Processing,* 26th Annual Meeting of the Association for Computational Linguistics, pp. 136–143.
7. Cutting, D., Kupiec, J., Pederson, J. & Sibun, P. (1992) in *Proceedings of the Third Conference on Applied Natural Language Processing, ACL*.
8. Hindle, D. (1989) in *Proceedings of the 27th Annual Meeting of the Association for Computational Linguistics*.
9. Merialdo, B. (1991) in *Proc. ICASSP-91*, pp. 809–812.
10. Weischedel, R., Meteer, M., Schwartz, R., Ramshaw, L. & Palmucci, J. (1993) in *Comput. Linguist.* **19,** 359–382.
11. Jelinek, F. (1995) *Proc. Natl. Acad. Sci. USA* **92,** 9964–9969.
12. Francis, W. N. (1964) Report to the U.S Office of Education on Cooperative Research Project No. E-007 (Brown Univ., Providence, RI).
13. Francis, W. N. & Kucera, H. (1982) *Lexicon and Grammar* (Houghton Mifflin, Boston).
14. Baker, J. K. (1979) in *Proceedings of the Spring Conference of the Acoustical Society of America*.
15. Jelinek, F., Lafferty, J. D. & Mercer, R. L. (1991) Continuous Speech Recognition Group (IBM T. J. Watson Research Center, Yorktown Heights, NY).
16. Lari, K. & Young, S. J. (1990) *Comput. Speech Lang.* **4,** 35–56.
17. Fujisaki, T., Jelinek, F., Cocke, J., Black, E. & Nishino, T. (1989) in *Proceedings of the First International Workshop on Parsing Technologies* (Carnegie–Mellon Univ., Pittsburgh).
18. Sharman, R. A., Jelinek, F. & Mercer, R. (1990) in *Proceedings of the Third DARPA Speech and Natural Language Workshop,* February.
19. Joshi, A. & Schabes, Y. (1992) in *Tree Automata and Languages,* eds. Nivat, M. & Podelski A. (Elsevier, New York).
20. Steedman, M. (1993) *Lingua* **90,** 221–258.
21. Pereira, F. & Schabes, Y. (1992) in *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*.
22. Brill, E., Magerman, D., Marcus, M. & Santorini, B. (1990) in *Proceedings of the DARPA Speech and Natural Language Workshop*, June, pp. 275–282.
23. Marcus, M., Santorini, B. & Marcinkiewicz, M. A. (1993) *Comput. Linguist.* **19,** 313–330.
24. Hemphill, C., Godfrey, J. & Doddington, G. (1990) in *Proceedings of the Third DARPA Speech and Natural Language Workshop*, February.
25. Black, E., Abney, S., Flickenger, F., Grishman, R., Harrison, P., Hindle, D., Ingria, R., Jelinek, F., Klavans, J., Liberman, M., Marcus, M., Roukos, S., Santorini, B. & Strzalkowski, T. (1991) in *Proceedings of the Fourth DARPA Speech and Natural Language Workshop*, February.
26. Magerman, D. & Marcus, M. (1991) in *Proceedings of the Fourth DARPA Speech and Natural Language Workshop*, February.
27. Magerman, D. & Marcus, M. (1991) in *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Berlin, April.
28. Zue, V., Glass, J., Goodine, D., Leung, H., McCandless, M., Philips, M., Polifroni, J. & Seneff, S. (1990) in *Proceedings of the Third DARPA Speech and Natural Language Workshop,* June.
29. Black, E., Lafferty, J. & Roukos, S. (1993) in *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics*.
30. Hindle, D. & Rooth, M. (1993) *Comput. Linguist.* **19,** 103–120.
31. Garside, R., Leech, G. & Sampson, G. (1987) *The Computational Analysis of English: A Corpus-Based Approach* (Longman, London).
32. Church, K., Gale, W., Hanks, P. & Hindle, D. (1991) *AT&T Bell Laboratories Technical Memorandum*.
33. Brown, P., Della Pietra, S., Della Pietra, V. & Mercer, R. (1991) in *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics* (Berkeley, CA).
34. Beckwith, R., Fellbaum, C., Gross, G. & Miller, G. (1991) in *Lexical Acquisition: Exploiting On-line Resources to Build a Lexicon*, ed. Zernik, U. (Erlbaum, Hillsdale, NJ), pp. 211–232.
35. Resnik, P. (1992) *Workshop Notes AAAI-92 Workshop in Statistically-Based NLP Techniques*, July.
36. Smadja, F. (1993) *Comput. Linguist.* **19,** 143–178.
37. Brent, M. (1993) *Comput. Linguist.* **19,** 243–262.
38. Pustojovsky, J., Bergler, S. & Anick, P. (1993) *Comput. Linguist.* **19,** 331–358.
39. Brill, E. (1993) in *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*.