Automatic Labeling of Semantic Roles

Daniel Gildea[*]
University of California, Berkeley, and
International Computer Science Institute

Daniel Jurafsky[†]
University of Colorado, Boulder

We present a system for identifying the semantic relationships, or ***semantic roles****, filled by constituents of a sentence within a semantic frame. Given an input sentence and a target word and frame, the system labels constituents with either abstract semantic roles, such as* AGENT *or* PATIENT*, or more domain-specific semantic roles, such as* SPEAKER, MESSAGE*, and* TOPIC.

The system is based on statistical classifiers trained on roughly 50,000 sentences that were hand-annotated with semantic roles by the FrameNet semantic labeling project. We then parsed each training sentence into a syntactic tree and extracted various lexical and syntactic features, including the phrase type of each constituent, its grammatical function, and its position in the sentence. These features were combined with knowledge of the predicate verb, noun, or adjective, as well as information such as the prior probabilities of various combinations of semantic roles. We used various lexical clustering algorithms to generalize across possible fillers of roles. Test sentences were parsed, were annotated with these features, and were then passed through the classifiers.

Our system achieves 82% accuracy in identifying the semantic role of presegmented constituents. At the more difficult task of simultaneously segmenting constituents and identifying their semantic role, the system achieved 65% precision and 61% recall.

Our study also allowed us to compare the usefulness of different features and feature combination methods in the semantic role labeling task. We also explore the integration of role labeling with statistical syntactic parsing and attempt to generalize to predicates unseen in the training data.

1. Introduction

Recent years have been exhilarating ones for natural language understanding. The excitement and rapid advances that had characterized other language-processing tasks such as speech recognition, part-of-speech tagging, and parsing have finally begun to appear in tasks in which understanding and semantics play a greater role. For example, there has been widespread commercial deployment of simple speech-based natural language understanding systems that answer questions about flight arrival times, give directions, report on bank balances, or perform simple financial transactions. More sophisticated research systems generate concise summaries of news articles, answer fact-based questions, and recognize complex semantic and dialogue structure.

But the challenges that lie ahead are still similar to the challenge that the field has faced since Winograd (1972): moving away from carefully hand-crafted, domain-dependent systems toward robustness and domain independence. This goal is not as

[*] Currently at Institute for Research in Cognitive Science, University of Pennsylvania, 3401 Walnut Street, Suite 400A, Philadelphia, PA 19104. E-mail: dgildea@cis.upenn.edu

[†] Departments of Linguistics and Computer Science, University of Colorado, Boulder, CO 80309. E-mail: jurafsky@colorado.edu

far away as it once was, thanks to the development of large semantic databases such as WordNet (Fellbaum 1998) and progress in domain-independent machine learning algorithms.

Current information extraction and dialogue understanding systems, however, are still based on domain-specific frame-and-slot templates. Systems for booking airplane information use domain-specific frames with slots like ORIG_CITY, DEST_CITY, or DEPART_TIME (Stallard 2000). Systems for studying mergers and acquisitions use slots like PRODUCTS, RELATIONSHIP, JOINT_VENTURE_COMPANY, and AMOUNT (Hobbs et al. 1997). For natural language understanding tasks to proceed beyond these specific domains, we need semantic frames and semantic understanding systems that do not require a new set of slots for each new application domain.

In this article we describe a shallow semantic interpreter based on semantic roles that are less domain specific than TO_AIRPORT or JOINT_VENTURE_COMPANY. These roles are defined at the level of semantic frames of the type introduced by Fillmore (1976), which describe abstract actions or relationships, along with their participants. For example, the JUDGEMENT frame contains roles like JUDGE, EVALUEE, and REASON, and the STATEMENT frame contains roles like SPEAKER, ADDRESSEE, and MESSAGE, as the following examples show:

(1) [$_{Judge}$ She] **blames** [$_{Evaluee}$ the Government] [$_{Reason}$ for failing to do enough to help] .

(2) [$_{Message}$ "I'll knock on your door at quarter to six"] [$_{Speaker}$ Susan] **said**.

These shallow semantic roles could play an important role in information extraction. For example, a semantic role parse would allow a system to realize that the *ruling* that is the direct object of *change* in (3) plays the same THEME role as the *ruling* that is the subject of *change* in (4):

(3) The canvassing board changed its ruling on Wednesday.

(4) The ruling changed because of the protests.

The fact that semantic roles are defined at the frame level means, for example, that the verbs *send* and *receive* would share the semantic roles (SENDER, RECIPIENT, GOODS, etc.) defined with respect to a common TRANSFER frame. Such common frames might allow a question-answering system to take a question like (5) and discover that (6) is relevant in constructing an answer to the question:

(5) Which party sent absentee ballots to voters?

(6) Both Democratic and Republican voters received absentee ballots from their party.

This shallow semantic level of interpretation has additional uses outside of generalizing information extraction, question answering, and semantic dialogue systems. One such application is in word sense disambiguation, where the roles associated with a word can be cues to its sense. For example, Lapata and Brew (1999) and others have shown that the different syntactic subcategorization frames of a verb such as *serve* can be used to help disambiguate a particular instance of the word. Adding semantic role subcategorization information to this syntactic information could extend this idea to use richer semantic knowledge. Semantic roles could also act as an important intermediate representation in statistical machine translation or automatic text summarization and in the emerging field of text data mining (TDM) (Hearst 1999). Finally, incorporating semantic roles into probabilistic models of language may eventually yield more accurate parsers and better language models for speech recognition.

This article describes an algorithm for identifying the semantic roles filled by constituents in a sentence. We apply statistical techniques that have been successful for the related problems of syntactic parsing, part-of-speech tagging, and word sense disambiguation, including probabilistic parsing and statistical classification. Our statistical algorithms are trained on a hand-labeled data set: the FrameNet database (Baker, Fillmore, and Lowe 1998; Johnson et al. 2001). The FrameNet database defines a tag set of semantic roles called **frame elements** and included, at the time of our experiments, roughly 50,000 sentences from the British National Corpus hand-labeled with these frame elements.

This article presents our system in stages, beginning in Section 2 with a more detailed description of the data and the set of frame elements or semantic roles used. We then introduce (in Section 3) the statistical classification technique used and examine in turn the knowledge sources of which our system makes use. Section 4 describes the basic syntactic and lexical features used by our system, which are derived from a Penn Treebank–style parse of individual sentences to be analyzed. We break our task into two subproblems: finding the relevant sentence constituents (deferred until Section 5), and giving them the correct semantic labels (Sections 4.2 and 4.3). Section 6 adds higher-level semantic knowledge to the system, attempting to model the selectional restrictions on role fillers not directly captured by lexical statistics. We compare hand-built and automatically derived resources for providing this information. Section 7 examines techniques for adding knowledge about systematic alternations in verb argument structure with sentence-level features. We combine syntactic parsing and semantic role identification into a single probability model in Section 8. Section 9 addresses the question of generalizing statistics from one target predicate to another, beginning with a look at domain-independent thematic roles in Section 9.1. Finally we draw conclusions and discuss future directions in Section 10.

2. Semantic Roles

Semantic roles are one of the oldest classes of constructs in linguistic theory, dating back thousands of years to Panini's *kāraka* theory (Misra 1966; Rocher 1964; Dahiya 1995). Longevity, in this case, begets variety, and the literature records scores of proposals for sets of semantic roles. These sets of roles range from the very specific to the very general, and many have been used in computational implementations of one type or another.

At the specific end of the spectrum are domain-specific roles such as the FROM_AIRPORT, TO_AIRPORT, or DEPART_TIME discussed above, or verb-specific roles such as EATER and EATEN for the verb *eat*. The opposite end of the spectrum consists of theories with only two "proto-roles" or "macroroles": PROTO-AGENT and PROTO-PATIENT (Van Valin 1993; Dowty 1991). In between lie many theories with approximately 10 roles, such as Fillmore's (1971) list of nine: AGENT, EXPERIENCER, INSTRUMENT, OBJECT, SOURCE, GOAL, LOCATION, TIME, and PATH.[1]

[1] There are scores of other theories with slightly different sets of roles, including those of Fillmore (1968), Jackendoff (1972), and Schank (1972); see Somers (1987) for an excellent summary.

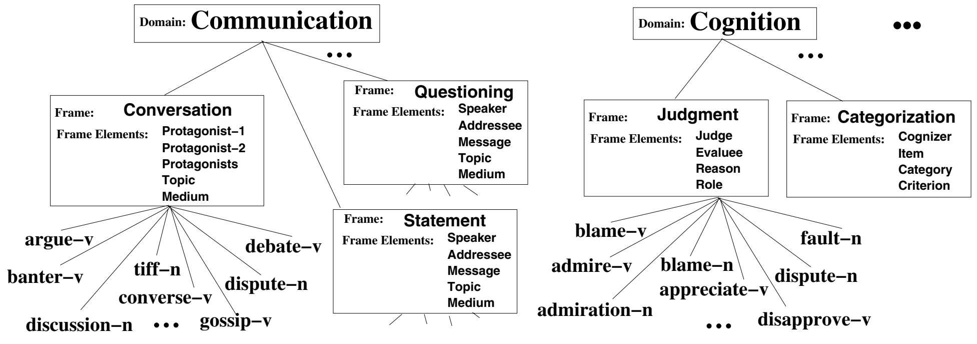

Figure 1
Sample domains and frames from the FrameNet lexicon.

Many of these sets of roles have been proposed by linguists as part of theories of **linking**, the part of grammatical theory that describes the relationship between semantic roles and their syntactic realization. Other sets have been used by computer scientists in implementing natural language understanding systems. As a rule, the more abstract roles have been proposed by linguists, who are more concerned with explaining generalizations across verbs in the syntactic realization of their arguments, whereas the more specific roles have more often been proposed by computer scientists, who are more concerned with the details of the realization of the arguments of specific verbs.

The FrameNet project (Baker, Fillmore, and Lowe 1998) proposes roles that are neither as general as the 10 abstract thematic roles, nor as specific as the thousands of potential verb-specific roles. FrameNet roles are defined for each semantic frame. A frame is a schematic representation of situations involving various participants, props, and other conceptual roles (Fillmore 1976). For example, the frame CONVERSATION, shown in Figure 1, is invoked by the semantically related verbs *argue*, *banter*, *debate*, *converse*, and *gossip*, as well as the nouns *dispute*, *discussion*, and *tiff*, and is defined as follows:

(7) Two (or more) people talk to one another. No person is construed as only a speaker or only an addressee. Rather, it is understood that both (or all) participants do some speaking and some listening: the process is understood to be symmetrical or reciprocal.

The roles defined for this frame, and shared by all its lexical entries, include PROTAGONIST-1 and PROTAGONIST-2 or simply PROTAGONISTS for the participants in the conversation, as well as MEDIUM and TOPIC. Similarly, the JUDGMENT frame mentioned above has the roles JUDGE, EVALUEE, and REASON and is invoked by verbs such as *blame*, *admire*, and *praise* and nouns such as *fault* and *admiration*. We refer to the roles for a given frame as frame elements. A number of hand-annotated examples from the JUDGMENT frame are included below to give a flavor of the FrameNet database:

(8) [$_{Judge}$ She] **blames** [$_{Evaluee}$ the Government] [$_{Reason}$ for failing to do enough to help] .

(9) Holman would characterise this as **blaming** [$_{Evaluee}$ the poor] .

(10) The letter quotes Black as saying that [$_{Judge}$ white and Navajo ranchers] misrepresent their livestock losses and **blame** [$_{Reason}$ everything] [$_{Evaluee}$ on coyotes] .

(11) The only dish she made that we could tolerate was [$_{Evaluee}$ syrup tart which[2]] [$_{Judge}$ we] **praised** extravagantly with the result that it became our unhealthy staple diet.

(12) I'm bound to say that I meet a lot of [$_{Judge}$ people who] **praise** [$_{Evaluee}$ me] [$_{Reason}$ for speaking up] but don't speak up themselves.

(13) Specimens of her verse translations of Tasso (*Jerusalem Delivered*) and Verri (*Roman Nights*) circulated to [$_{Manner}$ warm] [$_{Judge}$ critical] **praise**; but "unforeseen circumstance" prevented their publication.

(14) And if Sam Snort hails Doyler as monumental is he perhaps erring on the side of being excessive in [$_{Judge}$ his] **praise**?

Defining semantic roles at this intermediate frame level helps avoid some of the well-known difficulties of defining a unique small set of universal, abstract thematic roles while also allowing some generalization across the roles of different verbs, nouns, and adjectives, each of which adds semantics to the general frame or highlights a particular aspect of the frame. One way of thinking about traditional abstract thematic roles, such as Agent and Patient, in the context of FrameNet is to conceive them as frame elements defined by abstract frames, such as *action* and *motion*, at the top of an inheritance hierarchy of semantic frames (Fillmore and Baker 2000).

The examples above illustrate another difference between frame elements and thematic roles as commonly described in the literature. Whereas thematic roles tend to be arguments mainly of verbs, frame elements can be arguments of any predicate, and the FrameNet database thus includes nouns and adjectives as well as verbs.

The examples above also illustrate a few of the phenomena that make it hard to identify frame elements automatically. Many of these are caused by the fact that there is not always a direct correspondence between syntax and semantics. Whereas the subject of *blame* is often the Judge, the direct object of *blame* can be an Evaluee (e.g., *the poor* in "blaming the poor") or a Reason (e.g., *everything* in "blame everything on coyotes"). The identity of the Judge can also be expressed in a genitive pronoun, (e.g., *his* in "his praise") or even an adjective (e.g., *critical* in "critical praise").

The corpus used in this project is perhaps best described in terms of the methodology used by the FrameNet team. We outline the process here; for more detail see Johnson et al. (2001). As the first step, semantic frames were defined for the general domains chosen; the frame elements, or semantic roles for participants in a frame, were defined; and a list of **target words**, or lexical predicates whose meaning includes aspects of the frame, was compiled for each frame. Example sentences were chosen by searching the British National Corpus for instances of each target word. Separate searches were performed for various patterns over lexical items and part-of-speech sequences in the target words' context, producing a set of **subcorpora** for each target word, designed to capture different argument structures and ensure that some examples of each possible syntactic usage of the target word would be included in

[2] The FrameNet annotation includes both the relative pronoun and its antecedent in the target word's clause.

the final database. Thus, the focus of the project was on completeness of examples for lexicographic needs, rather than on statistically representative data. Sentences from each subcorpus were then annotated by hand, marking boundaries of each frame element expressed in the sentence and assigning tags for the annotated constituent's frame semantic role, syntactic category (e.g., noun phrase or prepositional phrase), and grammatical function in relation to the target word (e.g., object or complement of a verb). In the final phase of the process, the annotated sentences for each target word were checked for consistency. In addition to the tags just mentioned, the annotations include certain other information, which we do not make use of in this work, such as word sense tags for some target words and tags indicating metaphoric usages.

Tests of interannotator agreement were performed for data from a small number of predicates before the final consistency check. Interannotator agreement at the sentence level, including all frame element judgments and boundaries for one predicate, varied from .66 to .82 depending on the predicate. The kappa statistic (Siegel and Castellan 1988) varied from .67 to .82. Because of the large number of possible categories when boundary judgments are considered, kappa is nearly identical to the interannotator agreement. The system described in this article (which gets .65/.61 precision/recall on individual frame elements; see Table 15) correctly identifies all frame elements in 38% of test sentences. Although this .38 is not directly comparable to the .66–.82 interannotator agreements, it's clear that the performance of our system still falls significantly short of human performance on the task.

The British National Corpus was chosen as the basis of the FrameNet project despite differences between British and American usage because, at 100 million words, it provides the largest corpus of English with a balanced mixture of text genres. The British National Corpus includes automatically assigned syntactic part-of-speech tags for each word but does not include full syntactic parses. The FrameNet annotators did not make use of, or produce, a complete syntactic parse of the annotated sentences, although some syntactic information is provided by the grammatical function and phrase type tags of the annotated frame elements.

The preliminary version of the FrameNet corpus used for our experiments contained 67 frame types from 12 general semantic domains chosen for annotation. A complete list of the semantic domains represented in our data is shown in Table 1, along with representative frames and predicates. Within these frames, examples of a total of 1,462 distinct lexical predicates, or target words, were annotated: 927 verbs, 339 nouns, and 175 adjectives. There are a total of 49,013 annotated sentences and 99,232 annotated frame elements (which do not include the target words themselves).

How important is the particular set of semantic roles that underlies our system? For example, could the optimal choice of semantic roles be very dependent on the application that needs to exploit their information? Although there may well be application-specific constraints on semantic roles, our semantic role classifiers seem in practice to be relatively independent of the exact set of semantic roles under consideration. Section 9.1 describes an experiment in which we collapsed the FrameNet roles into a set of 18 abstract thematic roles. We then retrained our classifier and achieved roughly comparable results; overall performance was 82.1% for abstract thematic roles, compared to 80.4% for frame-specific roles. Although this doesn't show that the detailed set of semantic roles is irrelevant, it does suggest that our statistical classification algorithm, at least, is relatively robust to even quite large changes in role identities.

Table 1
Semantic domains with sample frames and predicates from the FrameNet lexicon.

Domain	Sample Frames	Sample Predicates
Body	Action	flutter, wink
Cognition	Awareness	attention, obvious
	Judgment	blame, judge
	Invention	coin, contrive
Communication	Conversation	bicker, confer
	Manner	lisp, rant
Emotion	Directed	angry, pleased
	Experiencer-Obj	bewitch, rile
General	Imitation	bogus, forge
Health	Response	allergic, susceptible
Motion	Arriving	enter, visit
	Filling	annoint, pack
Perception	Active	glance, savour
	Noise	snort, whine
Society	Leadership	emperor, sultan
Space	Adornment	cloak, line
Time	Duration	chronic, short
	Iteration	daily, sporadic
Transaction	Basic	buy, spend
	Wealthiness	broke, well-off

3. Related Work

Assignment of semantic roles is an important part of language understanding, and the problem of how to assign such roles has been attacked by many computational systems. Traditional parsing and understanding systems, including implementations of unification-based grammars such as Head-Driven Phrase Structure Grammar (HPSG) (Pollard and Sag 1994), rely on hand-developed grammars that must anticipate each way in which semantic roles may be realized syntactically. Writing such grammars is time consuming, and typically such systems have limited coverage.

Data-driven techniques have recently been applied to template-based semantic interpretation in limited domains by "shallow" systems that avoid complex feature structures and often perform only shallow syntactic analysis. For example, in the context of the Air Traveler Information System (ATIS) for spoken dialogue, Miller et al. (1996) computed the probability that a constituent such as *Atlanta* filled a semantic slot such as DESTINATION in a semantic frame for air travel. In a data-driven approach to information extraction, Riloff (1993) builds a dictionary of patterns for filling slots in a specific domain such as terrorist attacks, and Riloff and Schmelzenbach (1998) extend this technique to derive automatically entire "case frames" for words in the domain. These last systems make use of a limited amount of hand labor to accept or reject automatically generated hypotheses. They show promise for a more sophisticated approach to generalizing beyond the relatively small number of frames considered in the tasks. More recently, a domain-independent system has been trained by Blaheta and Charniak (2000) on the function tags, such as MANNER and TEMPORAL, included in the Penn Treebank corpus. Some of these tags correspond to FrameNet semantic roles, but the Treebank tags do not include all the arguments of most predicates. In this article, we aim to develop a statistical system for automatically learning to identify all semantic roles for a wide variety of predicates in unrestricted text.

4. Probability Estimation for Roles

In this section we describe the first, basic version of our statistically trained system for automatically identifying frame elements in text. The system will be extended in later sections. We first describe in detail the sentence- and constituent-level features on which our system is based and then use these features to calculate probabilities for predicting frame element labels in Section 4.2. In this section we give results for a system that labels roles using the human-annotated boundaries for the frame elements within the sentence; we return to the question of automatically identifying the boundaries in Section 5.

4.1 Features Used in Assigning Semantic Roles

Our system is a statistical one, based on training a classifier on a labeled training set and testing on a held-out portion of the data. The system is trained by first using an automatic syntactic parser to analyze the 36,995 training sentences, matching annotated frame elements to parse constituents and extracting various features from the string of words and the parse tree. During testing, the parser is run on the test sentences and the same features are extracted. Probabilities for each possible semantic role *r* are then computed from the features. The probability computation is described in the next section; here we discuss the features used.

The features used represent various aspects of the syntactic structure of the sentence as well as lexical information. The relationship between such surface manifestations and semantic roles is the subject of **linking theory** (see Levin and Rappaport Hovav [1996] for a synthesis of work in this area). In general, linking theory argues that the syntactic realization of arguments of a predicate is predictable from semantics; exactly how this relationship works, however, is the subject of much debate. Regardless of the underlying mechanisms used to generate syntax from semantics, the relationship between the two suggests that it may be possible to learn to recognize semantic relationships from syntactic cues, given examples with both types of information.

4.1.1 Phrase Type. Different semantic roles tend to be realized by different syntactic categories. For example, in communication frames, the SPEAKER is likely to appear as a noun phrase, TOPIC as a prepositional phrase or noun phrase, and MEDIUM as a prepositional phrase, as in: “[$_{Speaker}$ We] **talked** [$_{Topic}$ about the proposal] [$_{Medium}$ over the phone] .”

The phrase type feature we used indicates the syntactic category of the phrase expressing the semantic roles, using the set of syntactic categories of the Penn Treebank project, as described in Marcus, Santorini, and Marcinkiewicz (1993). In our data, frame elements are most commonly expressed as noun phrases (NPs, 47% of frame elements in the training set), and prepositional phrases (PPs, 22%). The next most common categories are adverbial phrases (ADVPs, 4%), particles (e.g. “make something *up*”; PRTs, 2%) and clauses (SBARs, 2%, and Ss, 2%). (Tables 22 and 23 in the Appendix provides a listing of Penn Treebank’s part-of-speech tags and constituent labels.)

We used Collins’ (1997) statistical parser trained on examples from the Penn Treebank to generate parses of the same format for the sentences in our data. Phrase types were derived automatically from parse trees generated by the parser, as shown in Figure 2. Given the automatically generated parse tree, the constituent spanning the same set of words as each annotated frame element was found, and the constituent’s nonterminal label was taken as the phrase type. In cases in which more than one constituent matches because of a unary production in the parse tree, the higher constituent was chosen.

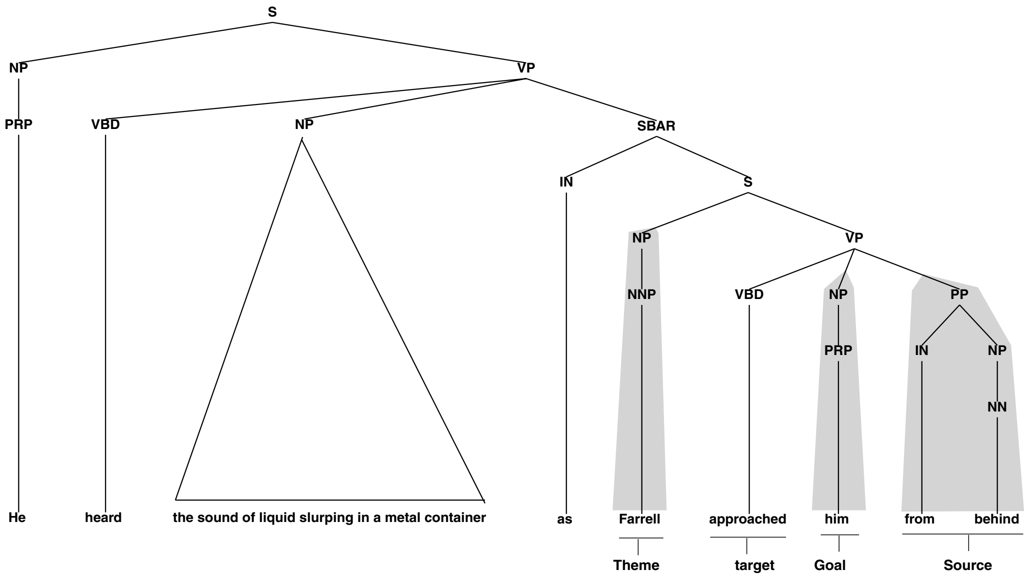

Figure 2
A sample sentence with parser output (above) and FrameNet annotation (below). Parse constituents corresponding to frame elements are highlighted.

The matching was performed by calculating the starting and ending word positions for each constituent in the parse tree, as well as for each annotated frame element, and matching each frame element with the parse constituent with the same beginning and ending points. Punctuation was ignored in this computation. Because of parsing errors, or, less frequently, mismatches between the parse tree formalism and the FrameNet annotation standards, for 13% of the frame elements in the training set, there was no parse constituent matching an annotated frame element. The one case of systematic mismatch between the parse tree formalism and the FrameNet annotation standards is the FrameNet convention of including both a relative pronoun and its antecedent in frame elements, as in the first frame element in the following sentence:

(15) In its rough state he showed it to [$_{Agt}$ the Professor, who] **bent** [$_{BPrt}$ his grey beard] [$_{Path}$ over the neat script] and read for some time in silence.

Mismatch caused by the treatment of relative pronouns accounts for 1% of the frame elements in the training set.

During testing, the largest constituent beginning at the frame element's left boundary and lying entirely within the element was used to calculate the frame element's features. We did not use this technique on the training set, as we expected that it would add noise to the data, but instead discarded examples with no matching parse constituent. Our technique for finding a near match handles common parse errors such as a prepositional phrase being incorrectly attached to a noun phrase at the right-hand edge, and it guarantees that some syntactic category will be returned: the part-of-speech tag of the frame element's first word in the limiting case.

4.1.2 Governing Category. The correlation between semantic roles and syntactic realization as subject or direct object is one of the primary facts that linking theory attempts to explain. It was a motivation for the case hierarchy of Fillmore (1968), which allowed such rules as "If there is an underlying AGENT, it becomes the syntactic subject." Similarly, in his theory of macroroles, Van Valin (1993) describes the ACTOR as being preferred in English for the subject. Functional grammarians consider syntactic subjects historically to have been grammaticalized agent markers. As an example of how such a feature can be useful, in the sentence "He drove the car over the cliff," the subject NP is more likely to fill the AGENT role than the other two NPs. We will discuss various grammatical-function features that attempt to indicate a constituent's syntactic relation to the rest of the sentence, for example, as a subject or object of a verb.

The first such feature, which we call "governing category," or *gov*, has only two values, S and VP, corresponding to subjects and objects of verbs, respectively. This feature is restricted to apply only to NPs, as it was found to have little effect on other phrase types. As with phrase type, the feature was read from parse trees returned by the parser. We follow links from child to parent up the parse tree from the constituent corresponding to a frame element until either an S or VP node is found and assign the value of the feature according to whether this node is an S or a VP. NP nodes found under S nodes are generally grammatical subjects, and NP nodes under VP nodes are generally objects. In most cases the S or VP node determining the value of this feature immediately dominates the NP node, but attachment errors by the parser or constructions such as conjunction of two NPs can cause intermediate nodes to be introduced. Searching for higher ancestor nodes makes the feature robust to such cases. Even given good parses, this feature is not perfect in discriminating grammatical functions, and in particular it confuses direct objects with adjunct NPs such as temporal phrases. For example, *town* in the sentence "He left town" and *yesterday* in the sentence "He left yesterday" will both be assigned a governing category of VP. Direct and indirect objects both appear directly under the VP node. For example, in the sentence "He gave me a new hose," *me* and *a new hose* are both assigned a governing category of VP. More sophisticated handling of such cases could improve our system.

4.1.3 Parse Tree Path. Like the governing-category feature described above, the parse tree path feature (*path*) is designed to capture the syntactic relation of a constituent to the rest of the sentence. The *path* feature, however, describes the syntactic relation between the target word (that is, the predicate invoking the semantic frame) and the constituent in question, whereas the *gov* feature is independent of where the target word appears in the sentence; that is, it identifies all subjects whether they are the subject of the target word or not.

The *path* feature is defined as the path from the target word through the parse tree to the constituent in question, represented as a string of parse tree nonterminals linked by symbols indicating upward or downward movement through the tree, as shown in Figure 3. Although the path is composed as a string of symbols, our system treats the string as an atomic value. The path includes, as the first element of the string, the part of speech of the target word and, as the last element, the phrase type or syntactic category of the sentence constituent marked as a frame element. After some experimentation, we settled on a version of the *path* feature that collapses the various part-of-speech tags for verbs, including past-tense verb (VBD), third-person singular present-tense verb (VBZ), other present-tense verb (VBP), and past participle (VBN), into a single verb tag denoted "VB."

Our *path* feature is dependent on the syntactic representation used, which in our case is the Treebank-2 annotation style (Marcus et al. 1994), as our parser is trained on this later version of the Treebank data. Figure 4 shows the annotation for the sentence "They expect him to cut costs throughout the organization," which exhibits

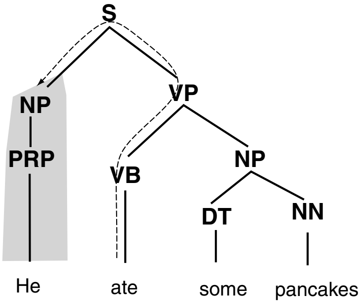

Figure 3
In this example, the path from the target word *ate* to the frame element *He* can be represented as VB↑VP↑S↓NP, with ↑ indicating upward movement in the parse tree and ↓ downward movement. The NP corresponding to *He* is found as described in Section 4.1.1.

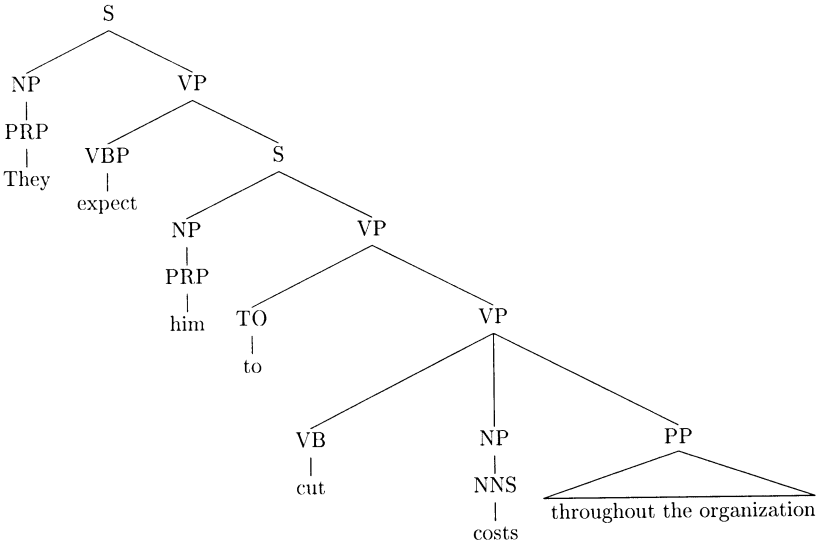

Figure 4
Treebank annotation of raising constructions.

the syntactic phenomenon known as subject-to-object raising, in which the main verb's object is interpreted as the embedded verb's subject. The Treebank-2 style tends to be generous in its usage of S nodes to indicate clauses, a decision intended to make possible a relatively straightforward mapping from S nodes to predications. In this example, the path from *cut* to the frame element *him* would be VB↑VP↑VP↑S↓NP, which typically indicates a verb's subject, despite the accusative case of the pronoun *him*. For the target word of *expect* in the sentence of Figure 4, the path to *him* would be VB↑VP↓S↓NP, rather than the typical direct-object path of VB↑VP↓NP.

An example of Treebank-2 annotation of an "equi" construction, in which a noun phrase serves as an argument of both the main and subordinate verbs, is shown in Figure 5. Here, an empty category is used in the subject position of the subordinate clause and is co-indexed with the NP *Congress* in the direct-object position of the main clause. The empty category, however, is not used in the statistical model of the parser or shown in its output and is also not used by the FrameNet annotation, which would mark the NP *Congress* as a frame element of *raise* in this example. Thus, the value of our *path* feature from the target word *raise* to the frame element *Congress* would

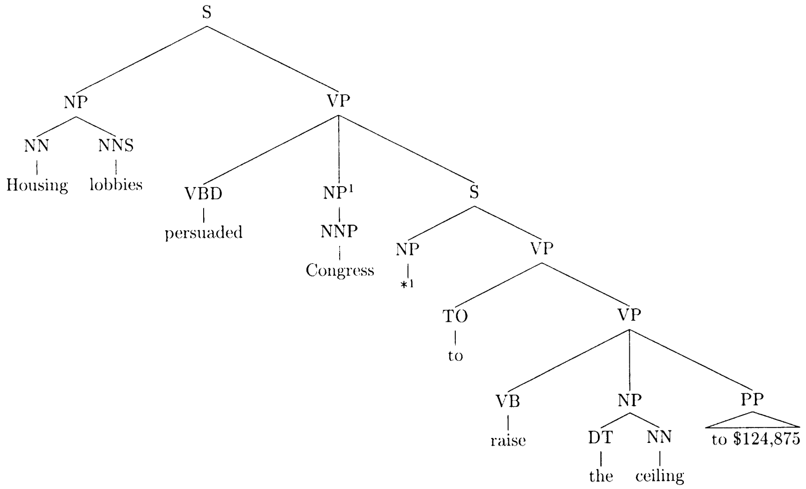

Figure 5
Treebank annotation of equi constructions. An empty category is indicated by an asterisk, and co-indexing by superscript numeral.

Table 2
Most frequent values of the path feature in the training data.

Frequency	Path	Description
14.2%	VB↑VP↓PP	PP argument/adjunct
11.8	VB↑VP↑S↓NP	Subject
10.1	VB↑VP↓NP	Object
7.9	VB↑VP↑VP↑S↓NP	Subject (embedded VP)
4.1	VB↑VP↓ADVP	Adverbial adjunct
3.0	NN↑NP↑NP↓PP	Prepositional complement of noun
1.7	VB↑VP↓PRT	Adverbial particle
1.6	VB↑VP↑VP↑VP↑S↓NP	Subject (embedded VP)
14.2		No matching parse constituent
31.4	Other	

be VB↑VP↑VP↑S↑VP↓NP, and from the target word of *persuaded* the path to *Congress* would be the standard direct-object path VB↑VP↓NP.

Other changes in annotation style from the original Treebank style were specifically intended to make predicate argument structure easy to read from the parse trees and include new empty (or null) constituents, co-indexing relations between nodes, and secondary functional tags such as *subject* and *temporal*. Our parser output, however, does not include this additional information, but rather simply gives trees of phrase type categories. The sentence in Figure 4 is one example of how the change in annotation style of Treebank-2 can affect this level of representation; the earlier style assigned the word *him* an NP node directly under the VP of *expect*.

The most common values of the *path* feature, along with interpretations, are shown in Table 2.

For the purposes of choosing a frame element label for a constituent, the *path* feature is similar to the *gov* feature defined above. Because the path captures more information than the governing category, it may be more susceptible to parser errors and data sparseness. As an indication of this, our *path* feature takes on a total of 2,978 possible values in the training data when frame elements with no matching

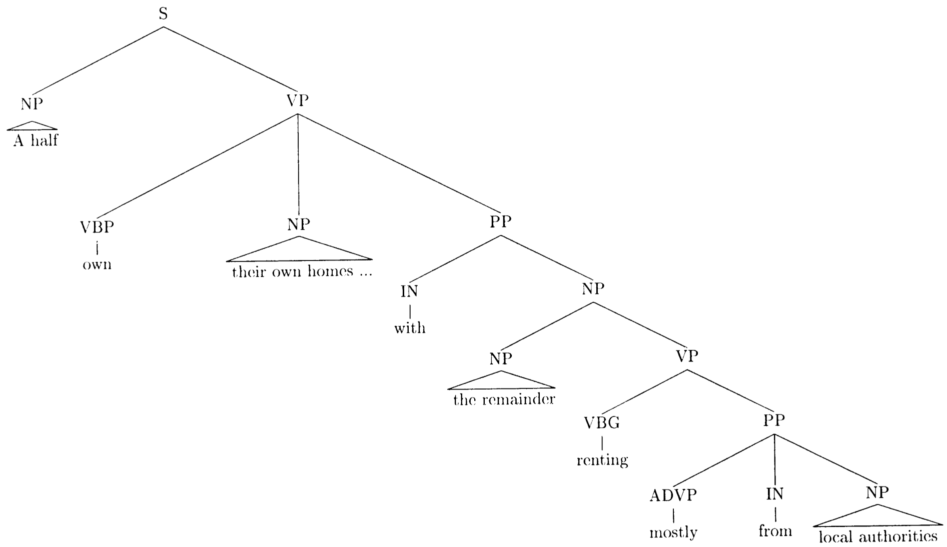

Figure 6
Example of target word *renting* in a small clause.

parse constituent are not counted and 4,086 possible values when paths are found to the best-matching constituent in these cases. The governing-category feature, on the other hand, which is defined only for NPs, has only two values (S, corresponding to subjects, and VP, corresponding to objects). In cases in which the *path* feature includes an S or VP ancestor of an NP node as part of the path to the target word, the *gov* feature is a function of the *path* feature. This is the case most of the time, including for our prototypical subject (VB↑VP↑S↓NP) and object (VB↑VP↓NP) paths. Of the 35,138 frame elements identified as NPs by the parser, only 4% have a *path* feature that does not include a VP or S ancestor. One such example is shown in Figure 6, where the small clause "the remainder renting ..." has no S node, giving a *path* feature from *renting* to *the remainder* of VB↑VP↑NP↓NP. The value of the *gov* feature here is VP, as the algorithm finds the VP of the sentence's main clause as it follows parent links up the tree. The feature is spurious in this case, because the main VP is not headed by, or relevant to, the target word *renting*.

Systems based on the *path* and *gov* features are compared in Section 4.3. The differences between the two are relatively small for the purpose of identifying semantic roles when frame element boundaries are known. The *path* feature will, however, be important in identifying which constituents are frame elements for a given target word, as it gives us a way of navigating through the parse tree to find the frame elements in the sentence.

4.1.4 Position. To overcome errors due to incorrect parses, as well as to see how much can be done without parse trees, we introduced *position* as a feature. The *position* feature simply indicates whether the constituent to be labeled occurs before or after the predicate defining the semantic frame. We expected this feature to be highly correlated with grammatical function, since subjects will generally appear before a verb and objects after.

Although we do not have hand-checked parses against which to measure the performance of the automatic parser on our corpus, the result that 13% of frame elements have no matching parse constituent gives a rough idea of the parser's accuracy. Almost all of these cases in which no matching parse constituent was found are due to parser error. Other parser errors include cases in which a constituent is found, but with the incorrect label or internal structure. This result also considers only the individual constituent representing the frame element: the parse for the rest of the sentence may be incorrect, resulting in an incorrect value for the grammatical function features described in the previous two sections. Collins (1997) reports 88% labeled precision and recall on individual parse constituents on data from the Penn Treebank, roughly consistent with our finding of at least 13% error.

4.1.5 Voice. The distinction between active and passive verbs plays an important role in the connection between semantic role and grammatical function, since direct objects of active verbs often correspond in semantic role to subjects of passive verbs. From the parser output, verbs were classified as active or passive by building a set of 10 passive-identifying patterns. Each of the patterns requires both a passive auxiliary (some form of *to be* or *to get*) and a past participle. Roughly 5% of the examples were identified as passive uses.

4.1.6 Head Word. As previously noted, we expected lexical dependencies to be extremely important in labeling semantic roles, as indicated by their importance in related tasks such as parsing. Head words of noun phrases can be used to express selectional restrictions on the semantic types of role fillers. For example, in a communication frame, noun phrases headed by *Bill*, *brother*, or *he* are more likely to be the SPEAKER, whereas those headed by *proposal*, *story*, or *question* are more likely to be the TOPIC. (We did not attempt to resolve pronoun references.)

Since the parser we used assigns each constituent a head word as an integral part of the parsing model, we were able to read the head words of the constituents from the parser output, employing the same set of rules for identifying the head child of each constituent in the parse tree. The rules for assigning a head word are listed in Collins (1999). Prepositions are considered to be the head words of prepositional phrases. The rules for assigning head words do not attempt to distinguish between cases in which the preposition expresses the semantic content of a role filler, such as PATH frame elements expressed by prepositional phrases headed by *along*, *through*, or *in*, and cases in which the preposition might be considered to be purely a case marker, as in most uses of *of*, where the semantic content of the role filler is expressed by the preposition's object. Complementizers are considered to be heads, meaning that infinitive verb phrases are always headed by *to* and subordinate clauses such as in the sentence "I'm sure that he came" are headed by *that*.

4.2 Probability Estimation

For our experiments, we divided the FrameNet corpus as follows: one-tenth of the annotated sentences for each target word were reserved as a test set, and another one-tenth were set aside as a tuning set for developing our system. A few target words where fewer than 10 examples had been chosen for annotation were removed from the corpus. (Section 9 will discuss generalization to unseen predicates.) In our corpus, the average number of sentences per target word is only 34, and the number of sentences per frame is 732, both relatively small amounts of data on which to train frame element classifiers.

To label the semantic role of a constituent automatically, we wish to estimate a probability distribution indicating how likely the constituent is to fill each possible

Table 3
Distributions calculated for semantic role identification: r indicates semantic role, pt phrase type, gov grammatical function, h head word, and t target word, or predicate.

Distribution	Coverage	Accuracy	Performance
$P(r \mid t)$	100.0%	40.9%	40.9%
$P(r \mid pt, t)$	92.5	60.1	55.6
$P(r \mid pt, gov, t)$	92.0	66.6	61.3
$P(r \mid pt, position, voice)$	98.8	57.1	56.4
$P(r \mid pt, position, voice, t)$	90.8	70.1	63.7
$P(r \mid h)$	80.3	73.6	59.1
$P(r \mid h, t)$	56.0	86.6	48.5
$P(r \mid h, pt, t)$	50.1	87.4	43.8

role, given the features described above and the predicate, or target word, t:

$$P(r \mid h, pt, gov, position, voice, t)$$

where r indicates semantic role, h head word, and pt phrase type. It would be possible to calculate this distribution directly from the training data by counting the number of times each role appears with a combination of features and dividing by the total number of times the combination of features appears:

$$P(r \mid h, pt, gov, position, voice, t) = \frac{\#(r, h, pt, gov, position, voice, t)}{\#(h, pt, gov, position, voice, t)}$$

In many cases, however, we will never have seen a particular combination of features in the training data, and in others we will have seen the combination only a small number of times, providing a poor estimate of the probability. The small number of training sentences for each target word and the large number of values that the head word feature in particular can take (any word in the language) contribute to the sparsity of the data. Although we expect our features to interact in various ways, we cannot train directly on the full feature set. For this reason, we built our classifier by combining probabilities from distributions conditioned on a variety of subsets of the features.

Table 3 shows the probability distributions used in the final version of the system. *Coverage* indicates the percentage of the test data for which the conditioning event had been seen in training data. *Accuracy* is the proportion of covered test data for which the correct role is given the highest probability, and *Performance*, which is the product of coverage and accuracy, is the overall percentage of test data for which the correct role is predicted.[3] Accuracy is somewhat similar to the familiar metric of *precision* in that it is calculated over cases for which a decision is made, and performance is similar to *recall* in that it is calculated over all true frame elements. Unlike in a traditional precision/recall trade-off, however, these results have no threshold to adjust, and the task is a multiway classification rather than a binary decision. The distributions calculated were simply the empirical distributions from the training data. That is, occurrences of each role and each set of conditioning events were counted in a table, and probabilities calculated by dividing the counts for each role by the total number

[3] Ties for the highest-probabilty role are resolved at random.

Table 4
Sample probabilities for $P(r \mid pt, gov, t)$ calculated from training data for the verb *abduct*. The variable *gov* is defined only for noun phrases. The roles defined for the *removing* frame in the *motion* domain are AGENT (AGT), THEME (THM), COTHEME (COTHM) ("... had been abducted *with him*"), and MANNER (MANR).

$P(r \mid pt, gov, t)$	**Count in training data**
$P(r = \textsc{Agt} \mid pt = \text{NP}, gov = \text{S}, t = \text{abduct}) = .46$	6
$P(r = \textsc{Thm} \mid pt = \text{NP}, gov = \text{S}, t = \text{abduct}) = .54$	7
$P(r = \textsc{Thm} \mid pt = \text{NP}, gov = \text{VP}, t = \text{abduct}) = 1$	9
$P(r = \textsc{Agt} \mid pt = \text{PP}, t = \text{abduct}) = .33$	1
$P(r = \textsc{Thm} \mid pt = \text{PP}, t = \text{abduct}) = .33$	1
$P(r = \textsc{CoThm} \mid pt = \text{PP}, t = \text{abduct}) = .33$	1
$P(r = \textsc{Manr} \mid pt = \text{ADVP}, t = \text{abduct}) = 1$	1

of observations for each conditioning event. For example, the distribution $P(r \mid pt, t)$ was calculated as follows:

$$P(r \mid pt, t) = \frac{\#(r, pt, t)}{\#(pt, t)}$$

Some sample probabilities calculated from the training are shown in Table 4.

As can be seen from Table 3, there is a trade-off between more-specific distributions, which have high accuracy but low coverage, and less-specific distributions, which have low accuracy but high coverage. The lexical head word statistics, in particular, are valuable when data are available but are particularly sparse because of the large number of possible head words.

To combine the strengths of the various distributions, we merged them in various ways to obtain an estimate of the full distribution $P(r \mid h, pt, gov, position, voice, t)$. The first combination method is linear interpolation, which simply averages the probabilities given by each of the distributions:

$$\begin{aligned} P(r \mid constituent) \quad = \quad & \lambda_1 P(r \mid t) + \lambda_2 P(r \mid pt, t) \\ & + \lambda_3 P(r \mid pt, gov, t) + \lambda_4 P(r \mid pt, position, voice) \\ & + \lambda_5 P(r \mid pt, position, voice, t) + \lambda_6 P(r \mid h) \\ & + \lambda_7 P(r \mid h, t) + \lambda_8 P(r \mid h, pt, t) \end{aligned}$$

where $\sum_i \lambda_i = 1$. The geometric mean, when expressed in the log domain, is similar:

$$\begin{aligned} P(r \mid constituent) \quad = \tfrac{1}{Z} \exp\{ \quad & \lambda_1 \log P(r \mid t) + \lambda_2 \log P(r \mid pt, t) \\ & + \lambda_3 \log P(r \mid pt, gov, t) + \lambda_4 \log P(r \mid pt, position, voice) \\ & + \lambda_5 \log P(r \mid pt, position, voice, t) + \lambda_6 \log P(r \mid h) \\ & + \lambda_7 \log P(r \mid h, t) + \lambda_8 \log P(r \mid h, pt, t) \quad \} \end{aligned}$$

where Z is a normalizing constant ensuring that $\sum_r P(r \mid constituent) = 1$.

Results for systems based on linear interpolation are shown in the first row of Table 5. These results were obtained using equal values of λ for each distribution defined for the relevant conditioning event (but excluding distributions for which the conditioning event was not seen in the training data). As a more sophisticated method of choosing interpolation weights, the expectation maximization (EM) algorithm was

Table 5
Results on development set, 8,167 observations.

Combining Method	Correct
Equal linear interpolation	79.5%
EM linear interpolation	79.3
Geometric mean	79.6
Backoff, linear interpolation	80.4
Backoff, geometric mean	79.6
Baseline: Most common role	40.9

Table 6
Results on test set, 7,900 observations.

Combining Method	Correct
EM linear interpolation	78.5%
Backoff, linear interpolation	76.9
Baseline: Most common role	40.6

used to estimate the likelihood of the observed role's being produced by each of the distributions in the general techniques of Jelinek and Mercer (1980). Because a number of the distributions used may have no training data for a given set of variables, the data were divided according to the set of distributions available, and a separate set of interpolation weights was trained for each set of distributions. This technique (line 2 of Table 5) did not outperform equal weights even on the data used to determine the weights. Although the EM algorithm is guaranteed to increase the likelihood of the training data, that likelihood does not always correspond to our scoring, which is based only on whether the correct outcome is assigned the highest probability. Results of the EM interpolation on held-out test data are shown in Table 6.

Experimentation has shown that the weights used have relatively little impact in our interpolation scheme, no doubt because the evaluation metric depends only on the ranking of the probabilities and not on their exact values. Changing the interpolation weights rarely changes the probabilities of the roles enough to change their ranking. What matters most is whether a combination of variables has been seen in the training data or not.

Results for the geometric mean are shown in row 3 of Table 5. As with linear interpolation, the exact weights were found to have little effect, and the results shown reflect equal weights. An area we have not explored is the use of the maximum-entropy techniques of, for example, Pietra, Pietra, and Lafferty (1997), to set weights for the log-linear model, either at the level of combining our probability distributions or at the level of calculating weights for individual values of the features.

In the "backoff" combination method, a lattice was constructed over the distributions in Table 3 from more-specific conditioning events to less-specific, as shown in Figure 7. The lattice is used to select a subset of the available distributions to combine. The less-specific distributions were used only when no data were present for any more-specific distribution. Thus, the distributions selected are arranged in a cut across the lattice representing the most-specific distributions for which data are available. The selected probabilities were combined with both linear interpolation and a geometric mean, with results shown in Table 5. The final row of the table represents the baseline

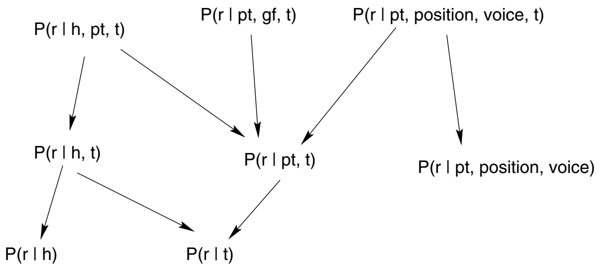

Figure 7
Lattice organization of the distributions from Table 3, with more-specific distributions toward the top.

of always selecting the most common role of the target word for all its constituents, that is, using only $P(r \mid t)$.

Although this lattice is reminiscent of techniques of backing off to less specific distributions commonly used in *n*-gram language modeling, it differs in that we use the lattice only to select distributions for which the conditioning event has been seen in the training data. Discounting and deleted interpolation methods in language modeling typically are used to assign small, nonzero probability to a predicted variable unseen in the training data even when a specific conditioning event has been seen. In our case, we are perfectly willing to assign zero probability to a specific role (the predicted variable). We are interested only in finding the role with the highest probability, and a role given a small, nonzero probability by smoothing techniques will still not be chosen as the classifier's output.

The lattice presented in Figure 7 represents just one way of choosing subsets of features for our system. Designing a feature lattice can be thought of as choosing a set of feature subsets: once the probability distributions of the lattice have been chosen, the graph structure of the lattice is determined by the subsumption relations among the sets of conditioning variables. Given a set of N conditioning variables, there are 2^N possible subsets, and 2^{2^N} possible sets of subsets, giving us a doubly exponential number of possible lattices. The particular lattice of Figure 7 was chosen to represent some expected interaction between features. For example, we expect *position* and *voice* to interact, and they are always used together. We expect the head word h and the phrase type pt to be relatively independent predictors of the semantic role and therefore include them separately as roots of the backoff structure. Although we will not explore all the possibilities for our lattice, some of the feature interactions are examined more closely in Section 4.3.

The final system performed at 80.4% accuracy, which can be compared to the 40.9% achieved by always choosing the most probable role for each target word, essentially chance performance on this task. Results for this system on test data, held out during development of the system, are shown in Table 6. Surprisingly, the EM-based interpolation performed better than the lattice-based system on the held-out test set, but not on the data used to set the weights in the EM-based system. We return to an analysis of which roles are hardest to classify in Section 9.1.

4.3 Interaction of Features

Three of our features, *position*, *gov*, and *path*, attempt to capture the syntactic relation between the target word and the constituent to be labeled, and in particular to differentiate the subjects from objects of verbs. To compare these three features directly, experiments were performed using each feature alone in an otherwise identical sys-

Table 7
Different estimators of grammatical function. The columns of the table correspond to Figures 8a, 8b, and 8c.

Feature	W/o voice	Independent voice feature	In conjunction with voice
path	79.4%	79.2%	80.4%
gov	79.1	79.2	80.7
position	79.9	79.7	80.5
—	76.3	76.0	76.0

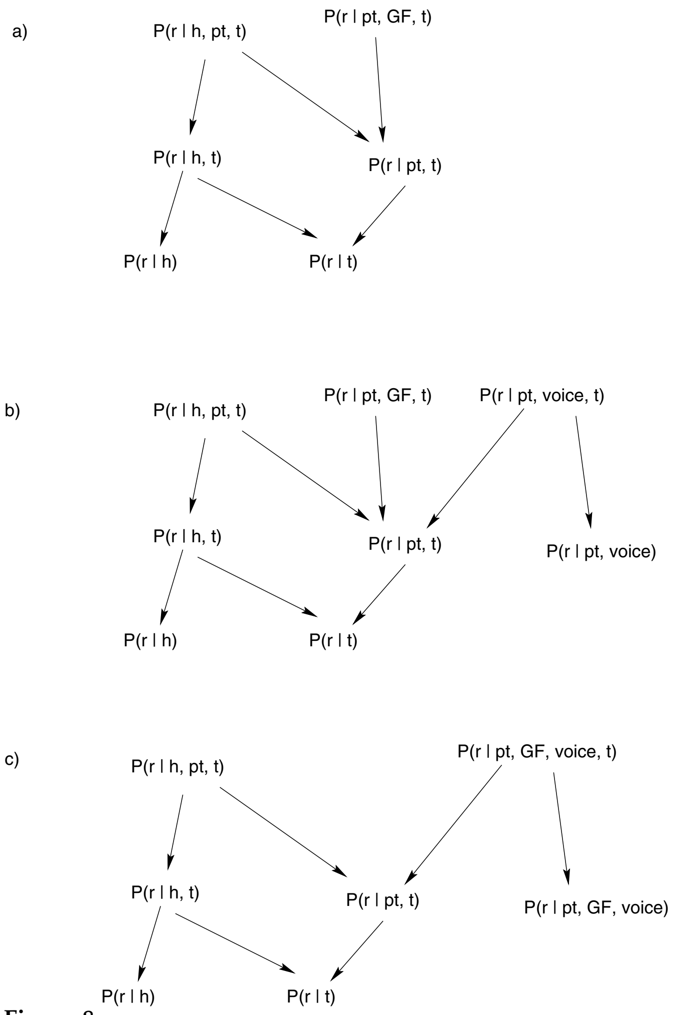

Figure 8
Lattice structures for comparing grammatical-function features.

tem. Results are shown in Table 7. For the first set of experiments, corresponding to the first column of Table 7, no voice information was used, with the result that the remaining distributions formed the lattice of Figure 8a. (“GF” (grammatical function) in the figure represents one of the features *position*, *gov*, and *path*.) Adding voice information back into the system independently of the grammatical-function feature results

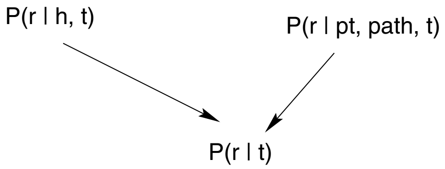

Figure 9
Minimal lattice.

in the lattice of Figure 8b, corresponding to the second column of Table 7. Choosing distributions such that the grammatical function and voice features are always used together results in Figure 8c, corresponding to the third column of Table 7. In each case, as in previous results, the grammatical function feature was used only when the candidate constituent was an NP. The last row of Table 7 shows results using no grammatical-function feature: the distributions making use of GF are removed from the lattices of Figure 8.

As a guideline for interpreting these results, with 8,167 observations, the threshold for statistical significance with $p < .05$ is a 1.0% absolute difference in performance. It is interesting to note that looking at a constituent's position relative to the target word performed as well as either of our features that read grammatical function off the parse tree, both with and without passive information. The *gov* and *path* features seem roughly equivalent in performance.

Using head word, phrase type, and target word without either position or grammatical function yielded only 76.3% accuracy, indicating that although the two features accomplish a similar goal, it is important to include some measure of the constituent's relationship to the target word, whether relative position or either of the syntactic features.

Use of the active/passive *voice* feature seems to be beneficial only when the feature is tied to grammatical function: the second column in Table 7 shows no improvement over the first, while the right-hand column, where grammatical function and voice are tied, shows gains (although only trends) of at least 0.5% in all cases. As before, our three indicators of grammatical function seem roughly equivalent, with the best result in this case being the *gov* feature. The lattice of Figure 8c performs as well as our system of Figure 7, indicating that including both position and either of the syntactic relations is redundant.

As an experiment to see how much can be accomplished with as simple a system as possible, we constructed the minimal lattice of Figure 9, which includes just two distributions, along with a prior for the target word to be used as a last resort when no data are available. This structure assumes that head word and grammatical function are independent. It further makes no use of the *voice* feature. We chose the *path* feature as the representation of grammatical function in this case. This system classified 76.3% of frame elements correctly, indicating that one can obtain roughly nine-tenths the performance of the full system with a simple approach. (We will return to a similar system for the purposes of cross-domain experiments in Section 9.)

5. Identification of Frame Element Boundaries

In this section we examine the system's performance on the task of locating the frame elements in a sentence. Although our probability model considers the question of finding the boundaries of frame elements separately from the question of finding the correct label for a particular frame element, similar features are used to calculate both

Table 8
Sample probabilities of a constituent's being a frame element.

Distribution	Sample Probability	Count in training data
$P(fe \mid path)$	$P(fe \mid path = \text{VB}\uparrow\text{VP}\downarrow\text{ADJP}\downarrow\text{ADVP}) = 1$	1
	$P(fe \mid path = \text{VB}\uparrow\text{VP}\downarrow\text{NP}) = .73$	3,963
	$P(fe \mid path = \text{VB}\uparrow\text{VP}\downarrow\text{NP}\downarrow\text{PP}\downarrow\text{S}) = 0$	22
$P(fe \mid path, t)$	$P(fe \mid path = \text{JJ}\uparrow\text{ADJP}\downarrow\text{PP}, t = \text{apparent}) = 1$	10
	$P(fe \mid path = \text{NN}\uparrow\text{NP}\uparrow\text{PP}\uparrow\text{VP}\downarrow\text{PP}, t = \text{departure}) = .4$	5
$P(fe \mid h, t)$	$P(fe \mid h = \text{sudden}, t = \text{apparent}) = 0$	2
	$P(fe \mid h = \text{to}, t = \text{apparent}) = .11$	93
	$P(fe \mid h = \text{that}, t = \text{apparent}) = .21$	81

probabilities. In the experiments below, the system is no longer given frame element boundaries but is still given as inputs the human-annotated target word and the frame to which it belongs. We do not address the task of identifying which frames come into play in a sentence but envision that existing word sense disambiguation techniques could be applied to the task.

As before, features are extracted from the sentence and its parse and are used to calculate probability tables, with the predicted variable in this case being *fe*, a binary indicator of whether a given constituent in the parse tree is or is not a frame element.

The features used were the *path* feature of Section 4.1.3, the identity of the target word, and the identity of the constituent's head word. The probability distributions calculated from the training data were $P(fe \mid path)$, $P(fe \mid path, t)$, and $P(fe \mid h, t)$, where *fe* indicates an event where the parse constituent in question is a frame element, *path* the path through the parse tree from the target word to the parse constituent, t the identity of the target word, and h the head word of the parse constituent. Some sample values from these distributions are shown in Table 8. For example, the path VB↑VP↓NP, which corresponds to the direct object of a verbal target word, had a high probability of being a frame element. The table also illustrates cases of sparse data for various feature combinations.

By varying the probability threshold at which a decision is made, one can plot a precision/recall curve as shown in Figure 10. $P(fe \mid path, t)$ performs relatively poorly because of fragmentation of the training data (recall that only about 30 sentences are available for each target word). Although the lexical statistic $P(fe \mid h, t)$ alone is not useful as a classifier, using it in linear interpolation with the path statistics improves results. The curve labeled "interpolation" in Figure 10 reflects a linear interpolation of the form

$$P(fe \mid p, h, t) = \lambda_1 P(fe \mid p) + \lambda_2 P(fe \mid p, t) + \lambda_3 P(fe \mid h, t) \tag{16}$$

Note that this method can identify only those frame elements that have a corresponding constituent in the automatically generated parse tree. For this reason, it is interesting to calculate how many true frame elements overlap with the results of the system, relaxing the criterion that the boundaries must match exactly. Results for partial matching are shown in Table 9. Three types of overlap are possible: the identified constituent entirely within the true frame element, the true frame element entirely within the identified constituent, and each sequence partially contained by the other. An example of the first case is shown in Figure 11, where the true MESSAGE frame element is *Mandarin by a head*, but because of an error in the parser output, no constituent exactly matches the frame element's boundaries. In this case, the system identifies

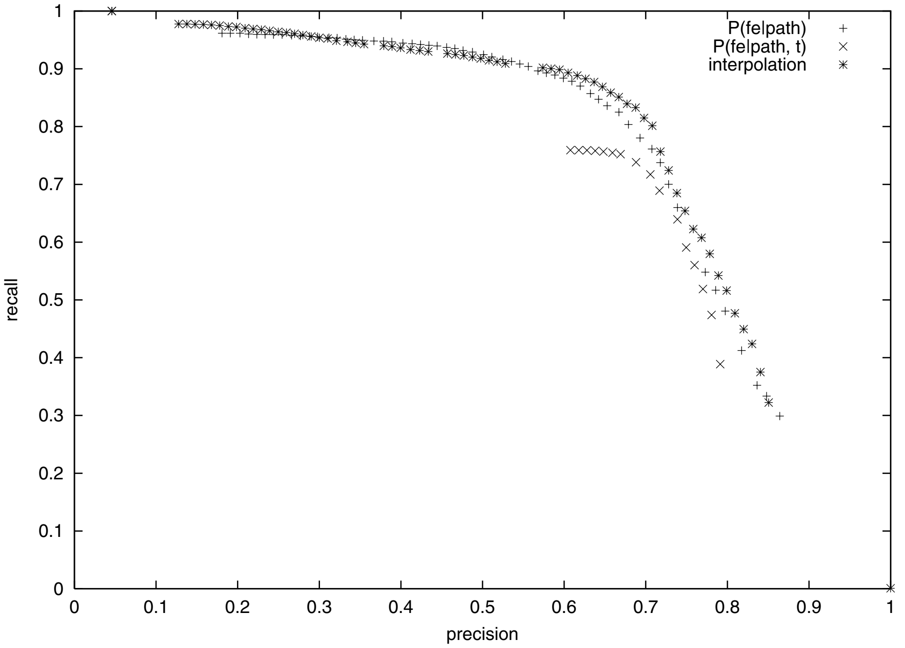

Figure 10
Plot of precision/recall curve for various methods of identifying frame elements. Recall is calculated over only frame elements with matching parse constituents.

Table 9
Results on identifying frame elements (FEs), including partial matches. Results obtained using *P*(*fe* | *path*) with threshold at 0.5. A total of 7,681 constituents were identified as FEs, and 8,167 FEs were present in hand-annotations, of which matching parse constituents were present for 7,053 (86%).

Type of Overlap	Identified Constituents	Number
Exactly matching boundaries	66%	5,421
Identified constituent entirely within true frame element	8	663
True frame element entirely within identified constituent	7	599
Both partially within the other	0	26
No overlap with any true frame element	13	972

two frame elements, indicated by shading, which together span the true frame element.

When the automatically identified constituents were fed through the role-labeling system described above, 79.6% of the constituents that had been correctly identified in the first stage were assigned the correct role in the second, roughly equivalent to the performance when roles were assigned to constituents identified by hand. A more sophisticated integrated system for identifying and labeling frame elements is described in Section 7.1.

6. Generalizing Lexical Statistics

As can be seen from Table 3, information about the head word of a constituent is valuable in predicting the constituent's role. Of all the distributions presented,

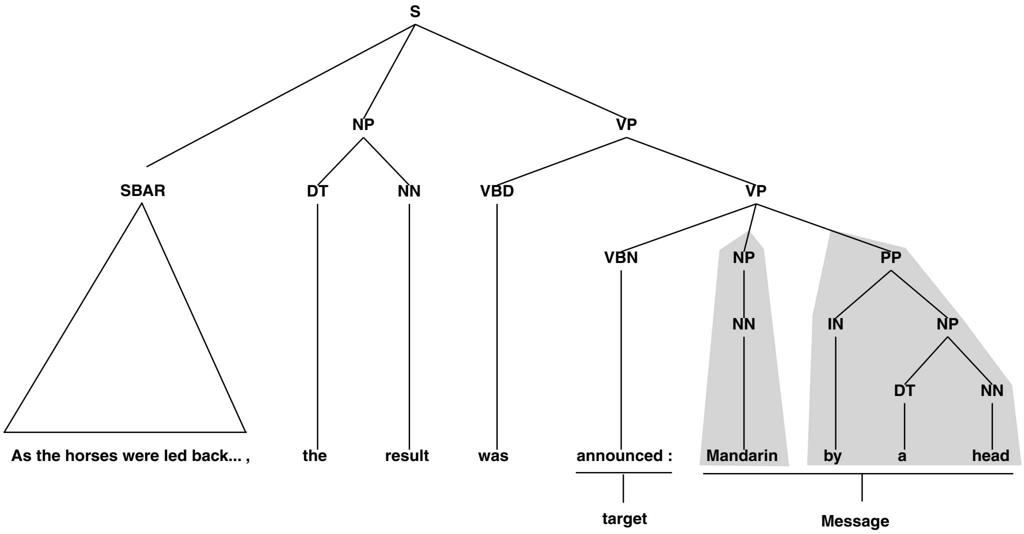

Figure 11
An example of overlap between identified frame elements and the true boundaries, caused by parser error. In this case two frame elements identified by the classifier (shaded subtrees) are entirely within the human annotation (indicated below the sentence), contributing two instances to row 2 of Table 9.

$P(r \mid h, pt, t)$ predicts the correct role most often (87.4% of the time) when training data for a particular head word have been seen. Because of the large vocabulary of possible head words, however, it also has the smallest coverage, meaning that it is likely that, for a given case in the test data, no frame element with the same head word will have been seen in the set of training sentences for the target word in question. To capitalize on the information provided by the head word, we wish to find a way to generalize from head words seen in the training data to other head words. In this section we compare three different approaches to the task of generalizing over head words: **automatic clustering** of a large vocabulary of head words to identify words with similar semantics; use of a hand-built ontological resource, WordNet, to organize head words in a **semantic hierarchy**; and **bootstrapping** to make use of unlabeled data in training the system. We will focus on frame elements filled by noun phrases, which constitute roughly half the total.

6.1 Automatic Clustering

To find groups of head words that are likely to fill the same semantic roles, an automatic clustering of nouns was performed using word co-occurrence data from a large corpus. This technique is based on the expectation that words with similar semantics will tend to co-occur with the same other sets of words. For example, nouns describing foods will tend to occur as direct objects of verbs such as *eat devour*, and *savor*. The clustering algorithm attempts to find such patterns of co-occurrence from the counts of grammatical relations between pairs of specific words in the corpus, without the use of any external knowledge or semantic representation.

We extracted verb–direct object relations from an automatically parsed version of the British National Corpus, using the parser of Carroll and Rooth (1998).[4] Clustering

[4] We are indebted to Mats Rooth and Sabine Schulte im Walde for providing us with the parsed corpus.

was performed using the probabilistic model of co-occurrence described in detail by Hofmann and Puzicha (1998). (For other natural language processing [NLP] applications of the probabilistic clustering algorithm, see, e.g., Rooth [1995], Rooth et al. [1999]; for application to language modeling, see Gildea and Hofmann [1999]. According to this model, the two observed variables, in this case the verb and the head noun of its object, can be considered independent given the value of a hidden cluster variable, c:

$$P(n, v) = \sum_{c} P(c)P(n \mid c)P(v \mid c)$$

One begins by setting a priori the number of values that c can take and using the EM algorithm to estimate the distributions $P(c)$, $P(n \mid c)$, and $P(v \mid c)$. Deterministic annealing was used to prevent overfitting of the training data.

We are interested only in the clusters of nouns given by the distribution $P(n \mid c)$: the verbs and the distribution $P(v \mid c)$ are thrown away once training is complete. Other grammatical relations besides direct object could be used, as could a set of relations. We used the direct object (following other clustering work such as Pereira, Tishby, and Lee [1993]) because it is particularly likely to exhibit semantically significant selectional restrictions.

A total of 2,610,946 verb-object pairs were used as training data for the clustering, with a further 290,105 pairs used as a cross-validation set to control the parameters of the clustering algorithm. Direct objects were identified as noun phrases directly under a verb phrase node—not a perfect technique, since it also finds nominal adjuncts such as "I start *today*." Forms of the verb *to be* were excluded from the data, as its co-occurrence patterns are not semantically informative. The number of values possible for the latent cluster variable was set to 256. (Comparable results were found with 64 clusters; the use of deterministic annealing prevents large numbers of clusters from resulting in overfitting.)

The soft clustering of nouns thus generated is used as follows: for each example in the frame element–annotated training data, probabilities for values of the hidden cluster variable were calculated using Bayes' rule:

$$P(c \mid h) = \frac{P(h \mid c)P(c)}{\sum_{c'} P(h \mid c')P(c')}$$

The clustering was applied only to noun phrase constituents; the distribution $P(n \mid c)$ from the clustering is used as a distribution $P(h \mid c)$ over noun head words.

Using the cluster probabilities, a new estimate of $P(r \mid c, pt, t)$ is calculated for cases where pt, the phrase type or syntactic category of the constituent, is NP:

$$P(r \mid c, pt, t) = \frac{\sum_{j:pt_j \;=\; pt, t_j \;=\; t, r_j \;=\; r} P(c_j \mid h_j)}{\sum_{j:pt_j \;=\; pt, t_j \;=\; t} P(c_j \mid h_j)}$$

where j is an index ranging over the frame elements in the training set and their associated features pt, t, h and their semantic roles r.

During testing, a smoothed estimate of the head word–based role probability is calculated by marginalizing over cluster values:

$$P(r \mid h, pt, t) = \sum_{c} P(r \mid c, pt, t)P(c \mid h)$$

again using $P(c \mid h) = \frac{P(h|c)P(c)}{\sum_{c'} P(h|c')P(c')}$.

Table 10
Clustering results on NP constituents only, 4,086 instances.

Distribution	Coverage	Accuracy	Performance
$P(r \mid h, pt, t)$	41.6%	87.0%	36.1%
$\sum_c P(r \mid c, pt, t)P(c \mid h)$	97.9	79.7	78.0
Interpolation of unclustered distributions	100.0	83.4	83.4
Unclustered distributions + clustering	100.0	85.0	85.0

As with the other methods of generalization described in this section, automatic clustering was applied only to noun phrases, which represent 50% of the constituents in the test data. We would not expect head word to be as valuable for other phrase types. The second most common category is prepositional phrases. The head of a prepositional phrase (PP) is considered to be the preposition, according to the rules we use, and because the set of prepositions is small, coverage is not as great a problem. Furthermore, the preposition is often a direct indicator of the semantic role. (A more complete model might distinguish between cases in which the preposition serves as a case or role marker and others in which it is semantically informative, with clustering performed on the preposition's object in the former case. We did not attempt to make this distinction.) Phrase types other than NP and PP make up only a small proportion of the data.

Table 10 shows results for the use of automatic clustering on constituents identified by the parser as noun phrases. As can be seen in the table, the vocabulary used for clustering includes almost all (97.9%) of the test data, and the decrease in accuracy from direct lexical statistics to clustered statistics is relatively small (from 87.0% to 79.7%). When combined with the full system described above, clustered statistics increase performance on NP constituents from 83.4% to 85.0% (statistically significant at $p < .05$). Over the entire test set, this translates into an improvement from 80.4% to 81.2%.

6.2 Using a Semantic Hierarchy: WordNet

The automatic clustering described above can be seen as an imperfect method of deriving semantic classes from the vocabulary, and we might expect a hand-developed set of classes to do better. We tested this hypothesis using WordNet (Fellbaum 1998), a freely available semantic hierarchy. The basic technique, when presented with a head word for which no training examples had been seen, was to ascend the type hierarchy until reaching a level for which training data are available. To do this, counts of training data were percolated up the semantic hierarchy in a technique similar to that of, for example, McCarthy (2000). For each training example, the count $\#(r, s, pt, t)$ was incremented in a table indexed by the semantic role r, WordNet sense s, phrase type pt, and target word t, for each WordNet sense s above the head word h in the hypernym hierarchy. In fact, the WordNet hierarchy is not a tree, but rather includes multiple inheritance. For example, *person* has as hypernyms both *life form* and *causal agent*. In such cases, we simply took the first hypernym listed, effectively converting the structure into a tree. A further complication is that several WordNet senses are possible for a given head word. We simply used the first sense listed for each word; a word sense disambiguation module capable of distinguishing WordNet senses might improve our results.

As with the clustering experiments reported above, the WordNet hierarchy was used only for noun phrases. The WordNet hierarchy does not include pronouns; to

Table 11
WordNet results on NP constituents only, 4,086 instances.

Distribution	Coverage	Accuracy	Performance
$P(r \mid h, pt, t)$	41.6%	87.0%	36.1%
WordNet: $P(r \mid s, pt, t)$	80.8	79.5	64.1
Interpolation of unclustered distributions	100.0	83.4	83.4
Unclustered distributions + WordNet	100.0	84.3	84.3

increase coverage, the personal pronouns *I*, *me*, *you*, *he*, *she*, *him*, *her*, *we*, and *us* were added as hyponyms of *person*. Pronouns that refer to inanimate, or both animate and inanimate, objects were not included. In addition, the CELEX English lexical database (Baayen, Piepenbrock, and Gulikers 1995) was used to convert plural nouns to their singular forms.

As shown in Table 11, accuracy for the WordNet technique is roughly the same as that in the automatic clustering results in Table 10: 84.3% on NPs, as opposed to 85.0% with automatic clustering. This indicates that the error introduced by the unsupervised clustering is roughly equivalent to the error caused by our arbitrary choice of the first WordNet sense for each word and the first hypernym for each WordNet sense. Coverage for the WordNet technique is lower, however, largely because of the absence of proper nouns from WordNet, as well as the absence of nonanimate pronouns (both personal pronouns such as *it* and *they* and indefinite pronouns such as *something* and *anyone*). A dictionary of proper nouns would likely help improve coverage, and a module for anaphora resolution might help cases with pronouns, with or without the use of WordNet. The conversion of plural forms to singular base forms was an important part of the success of the WordNet system, increasing coverage from 71.0% to 80.8%. Of the remaining 19.2% of all noun phrases not covered by the combination of lexical and WordNet sense statistics, 22% consisted of head words defined in WordNet, but for which no training data were available for any hypernym, and 78% consisted of head words not defined in WordNet.

6.3 Bootstrapping from Unannotated Data

A third way of attempting to improve coverage of the lexical statistics is to "bootstrap," or label unannotated data with the automatic system described in Sections 4 and 5 and use the (imperfect) result as further training data. This can be considered a variant of the EM algorithm, although we use the single most likely hypothesis for the unannotated data, rather than calculating the expectation over all hypotheses. Only one iteration of training on the unannotated data was performed.

The unannotated data used consisted of 156,590 sentences containing the target words of our corpus, increasing the total amount of data available to roughly six times the 36,995 annotated training sentences.

Table 12 shows results on noun phrases for the bootstrapping method. The accuracy of a system trained only on data from the automatic labeling (P_{auto}) is 81.0%, reasonably close to the 87.0% for the system trained only on annotated data (P_{train}). Combining the annotated and automatically labeled data increases coverage from 41.6% to 54.7% and performance to 44.5%. Because the automatically labeled data are not as accurate as the annotated data, we can do slightly better by using the automatic data only in cases where no training data are available, backing off to the distribution P_{auto} from P_{train}. The fourth row of Table 12 shows results with P_{auto} incorporated

Table 12
Bootstrapping results on NP constituents only, 4,086 instances.

Distribution	Coverage	Accuracy	Performance
$P_{train}(r \mid h, pt, t)$	41.6%	87.0%	36.1%
$P_{auto}(r \mid h, pt, t)$	48.2	81.0	39.0
$P_{train+auto}(r \mid h, pt, t)$	54.7	81.4	44.5
P_{train}, backoff to P_{auto}	54.7	81.7	44.7
Interpolation of unclustered distributions	100.0	83.4	83.4
Unclustered distributions + P_{auto}	100.0	83.2	83.2

into the backoff lattice of all the features of Figure 7, which actually resulted in a slight decrease in performance from the system without the bootstrapped data, shown in the third row. This is presumably because, although the system trained on automatically labeled data performed with reasonable accuracy, many of the cases it classifies correctly overlap with the training data. In fact our backing-off estimate of $P(r \mid h, pt, t)$ classifies correctly only 66% of the additional cases that it covers over $P_{train}(r \mid h, pt, t)$.

6.4 Discussion

The three methods of generalizing lexical statistics each had roughly equivalent accuracy on cases for which they were able to derive an estimate of the role probabilities for unseen head words. The differences between the three were primarily due to how much they could improve the *coverage* of the estimator, that is, how many new noun heads they were able to handle. The automatic-clustering method performed by far the best on this metric; only 2.1% of test cases were unseen in the data used for the automatic clustering. This indicates how much can be achieved with unsupervised methods given very large training corpora. The bootstrapping technique described here, although it has a similar unsupervised flavor, made use of much less data than the corpus used for noun clustering. Unlike probabilistic clustering, the bootstrapping technique can make use of only those sentences containing the target words in question. The WordNet experiment, on the other hand, indicates both the usefulness of hand-built resources when they apply and the difficulty of attaining broad coverage with such resources. Combining the three systems described would indicate whether their gains are complementary or overlapping.

7. Verb Argument Structure

One of the primary difficulties in labeling semantic roles is that one predicate may be used with different argument structures: for example, in the sentences "He opened the door" and "The door opened," the verb *open* assigns different semantic roles to its syntactic subject. In this section we compare two strategies for handling this type of alternation in our system: a sentence-level feature for frame element groups and a subcategorization feature for the syntactic uses of verbs. Then a simple system using the predicate's argument structure, or **syntactic signature**, as the primary feature will be contrasted with previous systems based on local, independent features.

7.1 Priors on Frame Element Groups

The system described in previous sections for classifying frame elements makes an important simplifying assumption: it classifies each frame element independent of the decisions made for the other frame elements in the sentence. In this section we remove

Table 13
Sample frame element groups for the verb *blame*.

Frame Element Group	Example Sentences
{Evaluee}	Holman would characterize this as **blaming** [$_{Evaluee}$ the poor] .
{Judge, Evaluee, Reason}	The letter quotes Black as saying that [$_{Judge}$ white and Navajo ranchers] misrepresent their livestock losses and **blame** [$_{Reason}$ everything] [$_{Evaluee}$ on coyotes] . [$_{Judge}$ She] **blames** [$_{Evaluee}$ the Government] [$_{Reason}$ for failing to do enough to help] .
{Judge, Evaluee}	The only dish she made that we could tolerate was [$_{Evaluee}$ syrup tart which] [$_{Judge}$ we] **praised** extravagantly with the result that it became our unhealthy staple diet.

Table 14
Frame element groups for the verb *blame* in the Judgment frame.

Frame Element Group	Probability
{Evaluee, Judge, Reason}	0.549
{Evaluee, Judge}	0.160
{Evaluee, Reason}	0.167
{Evaluee}	0.097
{Evaluee, Judge, Role }	0.014
{Judge}	0.007
{Judge, Reason}	0.007

this assumption and present a system that can make use of the information that, for example, a given target word requires that one role always be present or that having two instances of the same role is extremely unlikely.

To capture this information, we introduce the notion of a **frame element group**, which is the set of frame element roles present in a particular sentence (technically a multiset, as duplicates are possible, though quite rare). Frame element groups (FEGs) are *unordered*: examples are shown in Table 13. Sample probabilities from the training data for the frame element groups of the target word *blame* are shown in Table 14.

The FrameNet corpus recognizes three types of "null-instantiated" frame elements (Fillmore 1986), which are implied but do not appear in the sentence. An example of null instantiation is the sentence "Have you eaten?" where *food* is understood. We did not attempt to identify such null elements, and any null-instantiated roles are not included in the sentence's FEG. This increases the variability of observed FEGs, as a predicate may require a certain role but allow it to be null instantiated.

Our system for choosing the most likely overall assignment of roles for all the frame elements of a sentence uses an approximation that we derive beginning with the true probability of the optimal role assignment r^*:

$$r^* = \operatorname{argmax}_{r_{1\ldots n}} P(r_{1\ldots n} \mid t, f_{1\ldots n})$$

where $P(r_{1\ldots n} \mid t, f_{1\ldots n})$ represents the probability of an overall assignment of roles r_i to each of the n constituents of a sentence, given the target word t and the various features f_i of each of the constituents. In the first step we apply Bayes' rule to this quantity,

$$r^* = \operatorname{argmax}_{r_{1\ldots n}} P(r_{1\ldots n} \mid t) \frac{P(f_{1\ldots n} \mid r_{1\ldots n}, t)}{P(f_{1\ldots n} \mid t)}$$

and in the second we make the assumption that the features of the various constituents of a sentence are independent given the target word and each constituent's role and discard the term $P(f_{1\ldots n} \mid t)$, which is constant with respect to r:

$$r^* = \operatorname{argmax}_{r_{1\ldots n}} P(r_{1\ldots n} \mid t) \prod_i P(f_i \mid r_i, t)$$

We estimate the prior over frame element assignments as the probability of the frame element groups, represented with the set operator $\{\}$:

$$r^* = \operatorname{argmax}_{r_{1\ldots n}} P(\{r_{1\ldots n}\} \mid t) \prod_i P(f_i \mid r_i, t)$$

We then apply Bayes' rule again,

$$r^* = \operatorname{argmax}_{r_{1\ldots n}} P(\{r_{1\ldots n}\} \mid t) \prod_i \frac{P(r_i \mid f_i, t) P(f_i \mid t)}{P(r_i \mid t)}$$

and finally discard the feature prior $P(f_i \mid t)$ as being constant over the argmax expression:

$$r^* = \operatorname{argmax}_{r_{1\ldots n}} P(\{r_{1\ldots n}\} \mid t) \prod_i \frac{P(r_i \mid f_i, t)}{P(r_i \mid t)}$$

This leaves us with an expression in terms of the prior for frame element groups of a particular target word $P(\{r_{1\ldots n}\} \mid t)$, the local probability of a frame element given a constituent's features $P(r_i \mid f_i, t)$ on which our previous system was based, and the individual priors for the frame elements chosen $P(r_i \mid t)$. This formulation can be used to assign roles either when the frame element boundaries are known or when they are not, as we will discuss later in this section.

Calculating empirical FEG priors from the training data is relatively straightforward, but the sparseness of the data presents a problem. In fact, 15% of the test sentences had an FEG not seen in the training data for the target word in question. Using the empirical value for the FEG prior, these sentences could never be correctly classified. For this reason, we introduce a smoothed estimate of the FEG prior consisting of a linear interpolation of the empirical FEG prior and the product, for each possible frame element, of the probability of being present or not present in a sentence given the target word:

$$\lambda P(\{r_{1\ldots n}\} \mid t) + (1 - \lambda) \left[\prod_{r \in r_{1\ldots n}} P(r \in FEG \mid t) \prod_{r \notin r_{1\ldots n}} P(r \notin FEG \mid t) \right]$$

The value of λ was empirically set to maximize performance on the development set; a value of 0.6 yielded performance of 81.6%, a significant improvement over the 80.4% of the baseline system. Results were relatively insensitive to the exact value of λ.

Up to this point, we have considered separately the problems of labeling roles given that we know where the boundaries of the frame elements lie (Section 4, as well as Section 6) and finding the constituents to label in the sentence (Section 5).

Table 15
Combined results on boundary identification and role labeling.

	Unlabeled		Labeled	
Method	**Precision**	**Recall**	**Precision**	**Recall**
Boundary id. + baseline role labeler	72.6	63.1	67.0	46.8
Boundary id. + labeler w/FEG priors	72.6	63.1	65.9	46.2
Integrated boundary id. and labeling	74.0	70.1	64.6	61.2

We now turn to combining the two systems described above into a complete role labeling system. We use equation (16), repeated below, to estimate the probability that a constituent is a frame element:

$$P(fe \mid p, h, t) = \lambda_1 P(fe \mid p) + \lambda_2 P(fe \mid p, t) + \lambda_3 P(fe \mid h, t)$$

where p is the path through the parse tree from the target word to the constituent, t is the target word, and h is the constituent's head word.

The first two rows of Table 15 show the results when constituents are determined to be frame elements by setting the threshold on the probability $P(fe \mid p, h, t)$ to 0.5 and then running the labeling system of Section 4 on the resulting set of constituents. The first two columns of results show precision and recall for the task of identifying frame element boundaries correctly. The second pair of columns gives precision and recall for the combined task of boundary identification and role labeling; to be counted as correct, the frame element must both have the correct boundary and be labeled with the correct role.

Contrary to our results using human-annotated boundaries, incorporating FEG priors into the system based on automatically identified boundaries had a negative effect on labeled precision and recall. No doubt this is due to introducing a dependency on other frame element decisions that may be incorrect: the use of FEG priors causes errors in boundary identification to be compounded.

One way around this problem is to integrate boundary identification with role labeling, allowing the FEG priors and the role-labeling decisions to affect which constituents are frame elements. This was accomplished by extending the formulation

$$\text{argmax}_{r_{1\ldots n}} P(\{r_{1\ldots n}\} \mid t) \prod_i \frac{P(r_i \mid f_i, t)}{P(r_i \mid t)}$$

to include frame element identification decisions:

$$\text{argmax}_{r_{1\ldots n}} P(\{r_{1\ldots n}\} \mid t) \prod_i \frac{P(r_i \mid f_i, fe_i, t) P(fe_i \mid f_i)}{P(r_i \mid t)}$$

where fe_i is a binary variable indicating that a constituent is a frame element and $P(fe_i \mid f_i)$ is calculated as above. When fe_i is true, role probabilities are calculated as before; when fe_i is false, r_i assumes an empty role with probability one and is not included in the FEG represented by $\{r_{1\ldots n}\}$.

One caveat in using this integrated approach is its exponential complexity: each combination of role assignments to constituents is considered, and the number of combinations is exponential in the number of constituents. Although this did not pose a problem when only the annotated frame elements were under consideration, now we

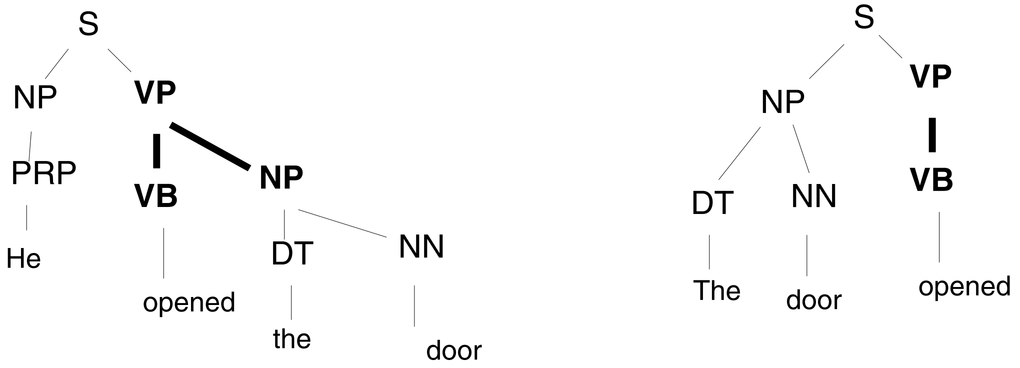

Figure 12
Two subcategorizations for the target word *open*. The relevant production in the parse tree is highlighted. On the left, the value of the feature is "VP → VB NP"; on the right it is "VP → VB."

must include every parse constituent with a nonzero probability for $P(fe_i \mid f_i)$. To make the computation tractable, we implement a pruning scheme: hypotheses are extended by choosing assignments for one constituent at a time, and only the top m hypotheses are retained for extension by assignments to the next constituent. Here we set $m = 10$ after experimentation showed that increasing m yielded no significant improvement.

Results for the integrated approach are shown in the last row of Table 15. Allowing role assignments to influence boundary identification improves results both on the unlabeled boundary identification task and on the combined identification and labeling task. The integrated approach puts us in a different portion of the precision/recall curve from the results in the first two rows, as it returns a higher number of frame elements (7,736 vs. 5,719). A more direct comparison can be made by lowering the probability threshold for frame element identification from 0.5 to 0.35 to force the nonintegrated system to return the same number of frame elements as the integrated system. This yields a frame element identification precision of 71.3% and recall of 67.6% and a labeled precision of 60.8% and recall of 57.6%, which is dominated by the result for the integrated system. The integrated system does not have a probability threshold to set; nonetheless it comes closer to identifying the correct number of frame elements (8,167) than does the independent boundary identifier when the theoretically optimal threshold of 0.5 is used with the latter.

7.2 Subcategorization

Recall that use of the FEG prior was motivated by the tendency of verbs to assign differing roles to the same syntactic position. For example, the verb *open* assigns different roles to the syntactic subject in *He opened the door* and *The door opened*. In this section we consider a different feature motivated by these problems: the syntactic subcategorization of the verb. For example, the verb *open* seems to be more likely to assign the role Patient to its subject in an intransitive context and Agent to its subject in a transitive context. Our use of a subcategorization feature was intended to differentiate between transitive and intransitive uses of a verb.

The feature used was the identity of the phrase structure rule expanding the target word's parent node in the parse tree, as shown in Figure 12. For example, for *He closed the door*, with *close* as the target word, the subcategorization feature would be "VP → VB NP." The subcategorization feature was used only when the target word was a verb. The various part-of-speech tags for verb forms (VBD for past-tense verb forms, VBZ for third-person singular present tense, VBP for other present tense, VBG for present participles, and VBN for past participles) were collapsed into a single tag VB. It is important to note that we are not able to distinguish complements from adjuncts, and our subcategorization feature could be sabotaged by cases such as *The door closed yesterday*. In the Penn Treebank style, *yesterday* is considered an NP with tree structure equivalent to that of a direct object. Our subcategorization feature is fairly specific: for example, the addition of an ADVP to a verb phrase will result in a different value. We tested variations of the feature that counted the number of NPs in a VP or the total number of children of the VP, with no significant change in results.

The subcategorization feature was used in conjunction with the *path* feature, which represents the sequence of nonterminals along the path through the parse tree from the target word to the constituent representing a frame element. Making use of the new subcategorization (*subcat*) feature by adding the distribution $P(r \mid subcat, path, t)$ to the lattice of distributions in the baseline system resulted in a slight improvement to 80.8% performance from 80.4%. As with the *gov* feature in the baseline system, it was found beneficial to use the *subcat* feature only for NP constituents.

7.3 Discussion

Combining the FEG priors and subcategorization feature into a single system resulted in performance of 81.6%, no improvement over using FEG priors without subcategorization. We suspect that the two seemingly different approaches in fact provide similar information. For example, in our hypothetical example of the sentence *He opened the door* vs. the sentence *The door opened*, the verb *open* would have high priors for the FEGs {AGENT, THEME} and {THEME}, but a low prior for {AGENT}. In sentences with only one candidate frame element (the subject in *The door closed*), the use of the FEG prior will cause it to be labeled THEME, even when the feature probabilities prefer labeling a subject as AGENT. Thus the FEG prior, by representing the set of arguments the predicate is likely to take, essentially already performs the function of the subcategorization feature.

The FEG prior allows us to introduce a dependency between the classifications of the sentence's various constituents with a single parameter. Thus, it can handle the alternation of our example without, for example, introducing the role chosen for one constituent as an additional feature in the probability distribution for the next constituent's role. It appears that because introducing additional features can further fragment our already sparse data, it is preferable to have a single parameter for the FEG prior.

An interesting result reinforcing this conclusion is that some of the argument-structure features that aided the system when individual frame elements were considered independently are unnecessary when using FEG priors. Removing the features *passive* and *position* from the system and using a smaller lattice of only the distributions not employing these features yields an improved performance of 82.8% on the role-labeling task using hand-annotated boundaries. We believe that, because these features pertain to syntactic alternations in how arguments are realized, they overlap with the function of the FEG prior. Adding unnecessary features to the system can reduce performance by fragmenting the training data.

8. Integrating Syntactic and Semantic Parsing

In the experiments reported in previous sections, we have used the parse tree returned by a statistical parser as input to the role-labeling system. In this section, we explore the interaction between semantic roles and syntactic parsing by integrating the parser with the semantic-role probability model. This allows the semantic-role assignment to affect the syntactic attachment decisions made by the parser, with the hope of improving the accuracy of the complete system.

Although most statistical parsing work measures performance in terms of syntactic trees without semantic information, an assignment of role fillers has been incorporated into a statistical parsing model by Miller et al. (2000) for the domain-specific templates of the Message Understanding Conference (Defense Advanced Research Projects Agency 1998) task. A key finding of Miller et al.'s work was that a system developed by annotating role fillers in text and training a statistical system performed at the same level as one based on writing a large system of rules, which requires much more highly skilled labor to design.

8.1 Incorporating Roles into the Parsing Model

We use as the baseline of all our parsing experiments the model described in Collins (1999). The algorithm is a form of **chart parsing**, which uses dynamic programming to search through the exponential number of possible parses by considering subtrees for each subsequence of the sentence independently. To apply chart parsing to a probabilistic grammar, independence relations must be assumed to hold between the probabilities of a parse tree and the internal structure of its subtrees.

In the case of stochastic context-free grammar, the probability of a tree is independent of the internal structure of its subtrees, given the topmost nonterminal of the subtree. The chart-parsing algorithm can simply find the highest-probability parse for each nonterminal for each substring of the input sentence. No lower-probability subtrees will ever be used in a complete parse, and they can be thrown away. Recent lexicalized stochastic parsers such as Collins (1999), Charniak (1997), and others add additional features to each constituent, the most important being the head word of the parse constituent.

The statistical system for assigning semantic roles described in the previous sections does not fit easily into the chart-parsing framework, as it relies on long-distance dependencies between the target word and its frame elements. In particular, the *path* feature, which is used to "navigate" through the sentence from the target word to its likely frame elements, may be an arbitrarily long sequence of syntactic constituents. A *path* feature looking for frame elements for a target word in another part of the sentence may examine the internal structure of a constituent, violating the independence assumptions of the chart parser. The use of priors over FEGs further complicates matters by introducing sentence-level features dependent on the entire parse.

For these reasons, we use the syntactic parsing model without frame element probabilities to generate a number of candidate parses, compute the best frame element assignment for each, and then choose the analysis with the highest overall probability. The frame element assignments are computed as in Section 7.1, with frame element probabilities being applied to every constituent in the parse.

To return a large number of candidate parses, the parser was modified to include constituents in the chart even when they were equivalent, according to the parsing model, to a higher-probability constituent. Rather than choosing a fixed n and keeping the n best constituents for each entry in the chart, we chose a probability threshold and kept all constituents within a margin of the highest-probability constituent. Thus the mechanism is similar to the beam search used to prune nonequivalent edges, but a lower threshold was used for equivalent edges ($\frac{1}{e}$ vs. $\frac{1}{100}$).

Using these pruning parameters, an average of 14.9 parses per sentence were obtained. After rescoring with frame element probabilities, 18% of the sentences were

Table 16
Results on rescoring parser output.

Method	Frame Element Precision	Frame Element Recall	Labeled Precision	Labeled Recall
Single-best parse	74.0	70.1	64.6	61.2
Rescoring parses	73.8	70.7	64.6	61.9

assigned a parse different from the original best parse. Nevertheless, the impact on identification of frame elements was small; results are shown in Table 16. The results show a slight, but not statistically significant, increase in recall of frame elements. One possible reason that the improvement is not greater is the relatively small number of parses per sentence available for rescoring. Unfortunately, the parsing algorithm used to generate n-best parses is inefficient, and generating large numbers of parses seems to be computationally intractable. In theory, the complexity of n-best variations of the Viterbi chart-parsing algorithm is quadratic in n. One can simply expand the dynamic programming chart to have n slots for the best solutions to each subproblem, rather than one. As our grammar forms new constituents from pairs of smaller constituents (that is, it internally uses a binarized grammar), for each pair of constituents considered in a single-best parser, up to n^2 pairs would be present in the n-best variant. The beam search used by modern parsers, however, makes the analysis more complex. Lexicalization of parse constituents dramatically increases the number of categories that must be stored in the chart, and efficient parsing requires that constituents below a particular probability threshold be dropped from further consideration. In practice, returning a larger number of parses with our algorithm seems to require increasing the pruning beam size to a degree that makes run times prohibitive.

In addition to the robustness of even relatively simple parsing models, one explanation for the modest improvement may be the fact that even our integrated system includes semantic information for only one word in the sentence. As the coverage of our frame descriptions increases, it may be possible to do better and to model the interactions between the frames invoked by a text.

9. Generalizing to Unseen Predicates

Most of the statistics used in the system as described above are conditioned on the target word, or predicate, for which semantic roles are being identified. This limits the applicability of the system to words for which training data are available. In Section 6, we attempted to generalize across fillers for the roles of a single predicate. In this section, we turn to the related but somewhat more difficult question of generalizing from seen to unseen predicates.

Many ways of attempting this generalization are possible, but the simplest is provided by the frame-semantic information of the FrameNet database. We can use data from target words in the same frame to predict behavior for an unseen word, or, if no data are available for the frame in question, we can use data from the same broad semantic domain into which the frames are grouped.

9.1 Thematic Roles

To investigate the degree to which our system is dependent on the set of semantic roles used, we performed experiments using abstract, general semantic roles such as

AGENT, PATIENT, and GOAL. Such roles were proposed in theories of linking such as Fillmore (1968) and Jackendoff (1972) to explain the syntactic realization of semantic arguments. This level of roles, often called **thematic roles**, was seen as useful for expressing generalizations such as "If a sentence has an AGENT, the AGENT will occupy the subject position." Such correlations might enable a statistical system to generalize from one semantic domain to another.

Recent work on linguistic theories of linking has attempted to explain syntactic realization in terms of the fundamentals of verbs' meaning (see Levin and Rappaport Hovav [1996] for a survey of a number of theories). Although such an explanation is desirable, our goal is more modest: an automatic procedure for identifying semantic roles in text. We aim to use abstract roles as a means of generalizing from limited training data in various semantic domains. We see this effort as consistent with various theoretical accounts of the underlying mechanisms of argument linking, since the various theories all postulate some sort of generalization between the roles of specific predicates.

To this end, we developed a correspondence from frame-specific roles to a set of abstract thematic roles. For each frame, an abstract thematic role was assigned to each frame element in the frame's definition. Since there is no canonical set of abstract semantic roles, we decided upon the list shown in Table 17. We are interested in adjuncts as well as arguments, leading to roles such as DEGREE not found in many theories of verb-argument linking. The difficulty of fitting many relations into standard categories such as AGENT and PATIENT led us to include other roles such as TOPIC. In all, we used 18 roles, a somewhat richer set than is often used, but still much more restricted than the frame-specific roles. Even with this enriched set, not all frame-specific roles fit neatly into one category.

An experiment was performed replacing each role tag in the training and test data with the corresponding thematic role and training the system as described above on the new dataset. Results were roughly comparable for the two types of semantic roles: overall performance was 82.1% for thematic roles, compared to 80.4% for frame-specific roles. This reflects the fact that most frames had a one-to-one mapping from frame-specific to abstract roles, so the tasks were largely equivalent. We expect abstract roles to be most useful when one is generalizing to predicates and frames not found in the training data, the topic of the following sections.

One interesting consequence of using abstract roles is that they allow us to compare more easily the system's performance on different roles because of the smaller number of categories. This breakdown is shown in Table 18. Results are given for two systems: the first assumes that the frame element boundaries are known and the second finds them automatically. The second system, which is described in Section 7.1, corresponds to the rightmost two columns in Table 18. The "Labeled Recall" column shows how often the frame element is correctly identified, whereas the "Unlabeled Recall" column shows how often a constituent with the given role is correctly identified as being a frame element, even if it is labeled with the wrong role.

EXPERIENCER and AGENT, two similar roles generally found as the subject for complementary sets of verbs, are the roles that are correctly identified the most often. The "Unlabeled Recall" column shows that these roles are easy to find in the sentence, as a predicate's subject is almost always a frame element, and the "Known Boundaries" column shows that they are also not often confused with other roles when it is known that they are frame elements. The two most difficult roles in terms of unlabeled recall, MANNER and DEGREE, are typically realized by adverbs or prepositional phrases and considered adjuncts. It is interesting to note that these are considered in FrameNet to be *general* frame elements that can be used in any frame.

Table 17
Abstract semantic roles, with representative examples from the FrameNet corpus.

Role	**Example**
AGENT	**Henry** *pushed* the door open and went in.
CAUSE	Jeez, **that** *amazes* me as well as riles me.
DEGREE	I **rather** *deplore* the recent manifestation of Pop; it doesn't seem to me to have the intellectual force of the art of the Sixties.
EXPERIENCER	It may even have been that **John** *anticipating* his imminent doom ratified some such arrangement perhaps in the ceremony at the Jordan.
FORCE	If this is the case can it be *substantiated* **by evidence from the history of developed societies**?
GOAL	Distant across the river the towers of the castle rose against the sky straddling the only land *approach* **into Shrewsbury**.
INSTRUMENT	In the children with colonic contractions **fasting motility** did not *differentiate* children with and without constipation.
LOCATION	These fleshy appendages are used to detect and *taste* food **amongst the weed and debris on the bottom of a river**.
MANNER	His brow *arched* **delicately**.
NULL	Yet while she had no intention of surrendering her home, **it** would be *foolish* to let the atmosphere between them become too acrimonious.
PATH	The dung-collector *ambled* slowly **over**, one eye on Sir John.
PATIENT	As soon as a character lays a hand on this item, the skeletal Cleric *grips* **it** more tightly.
PERCEPT	What is *apparent* is **that this manual is aimed at the non-specialist technician, possibly an embalmer who has good knowledge of some medical procedures**.
PROPOSITION	It says that rotation of partners does not *demonstrate* **independence**.
RESULT	All the arrangements for stay-behind agents in north-west Europe collapsed, but Dansey was able to *charm* most of the governments in exile in London **into recruiting spies**.
SOURCE	He heard the sound of liquid slurping in a metal container as Farrell *approached* him **from behind**.
STATE	Rex *spied* out Sam Maggott **hollering at all and sundry and making good use of his over-sized red gingham handkerchief**.
TOPIC	He said, "We would urge people to be aware and be *alert* **with fireworks** because your fun might be someone else's tragedy."

This section has shown that our system can use roles defined at a more abstract level than the corpus's frame-level roles and in fact that when we are looking at a single predicate, the choice has little effect. In the following sections, we attempt to use the abstract roles to generalize the behavior of semantically related predicates.

9.2 Unseen Predicates

We will present results at different, successively broader levels of generalization, making use of the categorization of FrameNet predicates into frames and more general semantic domains. We first turn to using data from the appropriate frame when no data for the target word are available.

Table 19 shows results for various probability distributions using a division of training and test data constructed such that no target words are in common. Every tenth target word was included in the test set. The amount of training data available for each frame varied, from just one target word in some cases to 167 target words in the "perception/noise" frame. The training set contained a total of 75,919 frame elements and the test set 7,801 frame elements.

Table 18
Performance broken down by abstract role. The third column represents accuracy when frame element boundaries are given to the system, and the fourth and fifth columns reflect finding the boundaries automatically. Unlabeled recall includes cases that were identified as a frame element but given the wrong role.

		Known Boundaries	Unknown Boundaries	
Role	**Number**	**% Correct**	**Labeled Recall**	**Unlabeled Recall**
Agent	2401	92.8	76.7	80.7
Experiencer	333	91.0	78.7	83.5
Source	503	87.3	67.4	74.2
Proposition	186	86.6	56.5	64.5
State	71	85.9	53.5	62.0
Patient	1161	83.3	63.1	69.1
Topic	244	82.4	64.3	72.1
Goal	694	82.1	60.2	69.6
Cause	424	76.2	61.6	73.8
Path	637	75.0	63.1	63.4
Manner	494	70.4	48.6	59.7
Percept	103	68.0	51.5	65.1
Degree	61	67.2	50.8	60.7
Null	55	65.5	70.9	85.5
Result	40	65.0	55.0	70.0
Location	275	63.3	47.6	63.6
Force	49	59.2	40.8	63.3
Instrument	30	43.3	30.0	73.3
(other)	406	57.9	40.9	63.1
Total	8167	82.1	63.6	72.1

Table 19
Cross-frame performance of various distributions. f represents the FrameNet semantic frame.

Distribution	Coverage	Accuracy	Performance
$P(r \mid path)$	95.3%	44.5%	42.4%
$P(r \mid path,f)$	87.4	68.7	60.1
$P(r \mid h)$	91.7	54.3	49.8
$P(r \mid h,f)$	74.1	81.3	60.3
$P(r \mid pt, position, voice)$	100.0	43.9	43.9
$P(r \mid pt, position, voice,f)$	98.7	68.3	67.4

The results show a familiar trade-off between coverage and accuracy. Conditioning both the head word and path features on the frame reduces coverage but improves accuracy. A linear interpolation,

$$\lambda_1 P(r \mid path,f) + \lambda_2 P(r \mid h,f) + \lambda_3 P(r \mid pt, position, voice,f)$$

achieved 79.4% performance on the test set, significantly better than any of the individual distributions and approaching the result of 82.1% for the original system, using target-specific statistics and thematic roles. This result indicates that predicates in the same frame behave similarly in terms of their argument structure, a finding generally consistent with theories of linking that claim that the syntactic realization of verb arguments can be predicted from their semantics. We would expect verbs in the same frame to be semantically similar and to have the same patterns of argument structure. The relatively high performance of frame-level statistics indicates that the

Table 20
Cross-frame performance of various distributions. *d* represents the FrameNet semantic domain.

Distribution	Coverage	Accuracy	Performance
$P(r \mid path)$	96.2%	41.2%	39.7%
$P(r \mid path, d)$	85.7	42.7	36.6
$P(r \mid h)$	91.0	44.7	40.6
$P(r \mid h, d)$	75.2	54.3	40.9
$P(r \mid d)$	95.1	29.9	28.4
$P(r)$	100.0	28.7	28.7

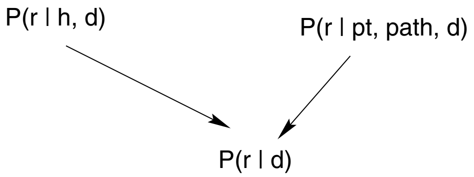

Figure 13
Minimal lattice for cross-frame generalization.

frames defined by FrameNet are fine-grained enough to capture the relevant semantic similarities.

This result is encouraging in that it indicates that a relatively small amount of data can be annotated for a few words in a semantic frame and used to train a system that can then bootstrap to a larger number of predicates.

9.3 Unseen Frames

More difficult than the question of unseen predicates in a known frame are frames for which no training data are present. The 67 frames in the current data set cover only a fraction of the English language, and the high cost of annotation makes it difficult to expand the data set to cover all semantic domains. The FrameNet project is defining additional frames and annotating data to expand the scope of the database. The question of how many frames exist, however, remains unanswered for the time being; a full account of frame semantics is expected to include multiple frames being invoked by many words, as well as an inheritance hierarchy of frames and a more detailed representation of each frame's meaning.

In this section, we examine the FrameNet data by holding out an entire frame for testing and using other frames from the same general semantic domain for training. Recall from Figure 1 that domains like COMMUNICATION include frames like CONVERSATION, QUESTIONING, and STATEMENT. Because of the variation in difficulty between different frames and the dependence of the results on which frames are held out for testing, we used a jackknifing methodology. Each frame was used in turn as test data, with all other frames used as training data. The results in Table 20 show average results over the entire data set.

Combining the distributions gives a system based on the (very restricted) backoff lattice of Figure 13. This system achieves performance of 51.0%, compared to 82.1% for the original system and 79.4% for the within-frame generalization task. The results show that generalizing across frames, even within a domain, is more difficult than generalizing across target words within a frame. There are several factors that may account for this: the FrameNet domains were intended primarily as a way of organizing the project, and their semantics have not been formalized. Thus, it may not be

Table 21
Cross-domain performance of various distributions.

Distribution	Coverage	Accuracy	Performance
$P(r \mid path)$	96.5%	35.3%	33.4%
$P(r \mid h)$	88.8	36.0	31.9
$P(r)$	100.0	28.7	28.7

surprising that they do not correspond to significant generalizations about argument structure. The domains are fairly broad, as indicated by the fact that always choosing the most common role for a given domain (the baseline for cross-frame, within-domain generalization, given as $P(r \mid d)$ in Table 20, classifies 28.4% of frame elements correctly) does not do better than the cross-domain baseline of always choosing the most common role from the entire database regardless of domain ($P(r)$ in Table 20, which yields 28.7% correct). This contrasts with a 40.9% baseline for $P(r \mid t)$, that is, always choosing the most common role for a particular target word (Table 5, last line). Domain information does not seem to help a great deal, given no information about the frame.

Furthermore, the cross-frame experiments here are dependent on the mapping of frame-level roles to abstract thematic roles. This mapping was done at the frame level; that is, FrameNet roles with the same label in two different frames may be translated into two different thematic roles, but all target words in the same frame make use of the same mapping. The mapping of roles within a frame is generally one to one, and therefore the choice of mapping has little effect when using statistics conditioned on the target word and on the frame, as in the previous section. When we are attempting to generalize between frames, the mapping determines which roles from the training frame are used to calculate probabilities for the roles in the test frames, and the choice of mapping is much more significant. The mapping used is necessarily somewhat arbitrary.

It is interesting to note that the path feature performs better when not conditioned on the domain. The head word, however, seems to be more domain-specific: although coverage declines when the context is restricted to the semantic domain, accuracy improves. This seems to indicate that the identity of certain role fillers is domain-specific, but that the syntax/semantics correspondence captured by the path feature is more general, as predicted by theories of syntactic linking.

9.4 Unseen Domains

As general as they are, the semantic domains of the current FrameNet database cover only a small portion of the language. The domains are defined at the level of, for example, COMMUNICATION and EMOTION; a list of the 12 domains in our corpus is given in Table 1. Whether generalization is possible across domains is an important question for a general language-understanding system.

For these experiments, a jackknifing protocol similar to that of the previous section was used, this time holding out one entire domain at a time and using all the others as training material. Results for the path and head word feature are shown in Table 21. The distributions $P(r \mid path)$, $P(r \mid h)$, and $P(r)$ of Table 21 also appeared in Table 20; the difference between the experiments is only in the division of training and test sets.

A linear interpolation, $\lambda_1 P(r \mid path) + \lambda_2 P(r \mid h)$, classifies 39.8% of frame elements correctly. This is no better than our result of 40.9% (Table 3) for always choosing a predicate's most frequent role; however, the cross-domain system does not have role frequencies for the test predicates.

9.5 Discussion

As one might expect, as we make successively broader generalizations to semantically more distant predicates, performance degrades. Our results indicate that frame semantics give us a level at which generalizations relevant to argument linking can be made. Our results for unseen predicates within the same frame are encouraging, indicating that the predicates are semantically similar in ways that result in similar argument structure, as the semantically based theories of linking advocated by Levin (1993) and Levin and Rappaport Hovav (1996) would predict. We hope that corpus-based systems such as ours can provide a way of testing and elaborating such theories in the future. We believe that some level of skeletal representation of the relevant aspects of a word's meaning, along the lines of Kipper et al. (2000) and of the frame hierarchy being developed by the FrameNet project, could be used in the future to help a statistical system generalize from similar words for which training data are available.

10. Conclusion

Our system is able to label semantic roles automatically with fairly high accuracy, indicating promise for applications in various natural language tasks. Semantic roles do not seem to be simple functions of a sentence's syntactic tree structure, and lexical statistics were found to be extremely valuable, as has been the case in other natural language processing applications. Although lexical statistics are quite accurate on the data covered by observations in the training set, the sparsity of their coverage led us to introduce semantically motivated knowledge sources, which in turn allowed us to compare automatically derived and hand-built semantic resources. Various methods of extending the coverage of lexical statistics indicated that the broader coverage of automatic clustering outweighed its imprecision. Carefully choosing sentence-level features for representing alternations in verb argument structure allowed us to introduce dependencies between frame element decisions within a sentence without adding too much complexity to the system. Integrating semantic interpretation and syntactic parsing yielded only the slightest gain, showing that although probabilistic models allow easy integration of modules, the gain over an unintegrated system may not be large because of the robustness of even simple probabilistic systems.

Many aspects of our system are still quite preliminary. For example, our system currently assumes knowledge of the correct frame type for the target word to determine the semantic roles of its arguments. A more complete semantic analysis system would thus require a module for frame disambiguation. It is not clear how difficult this problem is and how much it overlaps with the general problem of word-sense disambiguation.

Much else remains to be done to apply the system described here to the interpretation of general text. One technique for dealing with the sparseness of lexical statistics would be the combination of FrameNet data with named-entity systems for recognizing times, dates, and locations, the effort that has gone into recognizing these items, typically used as adjuncts, should complement the FrameNet data, which is more focused on arguments. Generalization to predicates for which no annotated data are available may be possible using other lexical resources or automatic clustering of predicates. Automatically learning generalizations about the semantics and syntactic behavior of predicates is an exciting problem for the years to come.

Appendix

Table 22
Penn Treebank part-of-speech tags (including punctuation).

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	*+,%, &*
CD	Cardinal number	*one, two, three*	TO	“to”	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential ‘there’	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past tense	*ate*
IN	Preposition/sub-conj	*of, in, by*	VBG	Verb, gerund	*eating*
JJ	Adjective	*yellow*	VBN	Verb, past participle	*eaten*
JJR	Adj., comparative	*bigger*	VBP	Verb, non-3sg pres	*eat*
JJS	Adj., superlative	*wildest*	VBZ	Verb, 3sg pres	*eats*
LS	List item marker	*1, 2, One*	WDT	Wh-determiner	*which, that*
MD	Modal	*can, should*	WP	Wh-pronoun	*what, who*
NN	Noun, sing. or mass	*llama*	WP$	Possessive wh-	*whose*
NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
NNP	Proper noun, singular	*IBM*	$	Dollar sign	$
NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	#
PDT	Predeterminer	*all, both*	“	Left quote	(‘ or “)
POS	Possessive ending	*’s*	”	Right quote	(’ or ”)
PRP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PRP$	Possessive pronoun	*your, one’s*	)	Right parenthesis	( ],), }, >)
RB	Adverb	*quickly, never*	,	Comma	,
RBR	Adverb, comparative	*faster*	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	*fastest*	:	Mid-sentence punc	(: ; ... — -)
RP	Particle	*up, off*			

Table 23
Penn Treebank constituent (or nonterminal) labels.

Label	Description
ADJP	Adjective Phrase
ADVP	Adverb Phrase
CONJP	Conjunction Phrase
FRAG	Fragment
INTJ	Interjection
NAC	Not a constituent
NP	Noun Phrase
NX	Head subphrase of complex noun phrase
PP	Prepositional Phrase
QP	Quantifier Phrase
RRC	Reduced Relative Clause
S	Simple declarative clause (sentence)
SBAR	Clause introduced by complementizer
SBARQ	Question introduced by wh-word
SINV	Inverted declarative sentence
SQ	Inverted yes/no question
UCP	Unlike Co-ordinated Phrase
VP	Verb Phrase
WHADJP	*Wh*-adjective Phrase
WHADVP	*Wh*-adverb Phrase
WHNP	*Wh*-noun Phrase
WHPP	*Wh*-prepositional Phrase

Acknowledgments

We are grateful to Chuck Fillmore, Andreas Stolcke, Jerry Feldman, and three anonymous reviewers for their comments and suggestions, to Collin Baker for his assistance with the FrameNet data, and to Mats Rooth and Sabine Schulte im Walde for making available their parsed corpus. This work was primarily funded by National Science Foundation grant ITR/HCI #0086132 to the FrameNet project.

References

Baayen, R. H., R. Piepenbrock, and L. Gulikers. 1995. *The CELEX Lexical Database (Release 2) [CD-ROM]*. Linguistic Data Consortium, University of Pennsylvania [Distributor], Philadelphia, PA.

Baker, Collin F., Charles J. Fillmore, and John B. Lowe. 1998. "The Berkeley FrameNet project." In *Proceedings of COLING/ACL*, pages 86–90, Montreal, Canada.

Blaheta, Don and Eugene Charniak. 2000. "Assigning function tags to parsed text." In *Proceedings of the First Annual Meeting of the North American Chapter of the ACL (NAACL)*, pages 234–240, Seattle, Washington.

Carroll, Glenn and Mats Rooth. 1998. "Valence induction with a head-lexicalized PCFG." In *Proceedings of the Third Conference on Empirical Methods in Natural Language Processing (EMNLP 3)*, Granada, Spain.

Charniak, Eugene. 1997. "Statistical parsing with a context-free grammar and word statistics." In *AAAI-97*, pages 598–603, Menlo Park, August. AAAI Press, Menlo Park, California.

Collins, Michael. 1997. "Three generative, lexicalised models for statistical parsing." In *Proceedings of the 35th Annual Meeting of the ACL*, pages 16–23, Madrid, Spain.

Collins, Michael. 1999. Head-Driven Statistical Models for Natural Language Parsing. Ph.D. dissertation, University of Pennsylvania, Philadelphia.

Dahiya, Yajan Veer. 1995. *Panini as a Linguist: Ideas and Patterns*. Eastern Book Linkers, Delhi, India.

Defense Advanced Research Projects Agency, editor. 1998. *Proceedings of the Seventh Message Understanding Conference*.

Dowty, David R. 1991. Thematic proto-roles and argument selection. *Language* 67(3):547–619.

Fellbaum, Christiane, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Massachusetts.

Fillmore, Charles J. 1968. "The case for case." In Emmon W. Bach and Robert T. Harms, editors, *Universals in Linguistic Theory*. Holt, Rinehart & Winston, New York, pages 1–88.

Fillmore, Charles J. 1971. "Some problems for case grammar." In R. J. O'Brien, editor, *22nd Annual Round Table. Linguistics: Developments of the Sixties—Viewpoints of the Seventies*. Volume 24 of *Monograph Series on Language and Linguistics*. Georgetown University Press, Washington, D.C., pages 35–56.

Fillmore, Charles J. 1976. "Frame semantics and the nature of language." In *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, Volume 280, pages 20–32. New York Academy of Sciences, New York.

Fillmore, Charles J. 1986. "Pragmatically controlled zero anaphora." In *Proceedings of Berkeley Linguistics Society*, pages 95–107, Berkeley, California.

Fillmore, Charles J. and Collin F. Baker. 2000. "FrameNet: Frame semantics meets the corpus." Poster presentation, 74th Annual Meeting of the Linguistics Society of America.

Gildea, Daniel and Thomas Hofmann. 1999. "Probabilistic topic analysis for language modeling." In *Eurospeech-99*, pages 2167–2170, Budapest.

Hearst, Marti. 1999. "Untangling text data mining." In *Proceedings of the 37th Annual Meeting of the ACL*, pages 3–10, College Park, Maryland.

Hobbs, Jerry R., Douglas Appelt, John Bear, David Israel, Megumi Kameyama, Mark E. Stickel, and Mabry Tyson. 1997. "FASTUS: A cascaded finite-state transducer for extracting information from natural-language text." In Emmanuel Roche and Yves Schabes, editors, *Finite-State Language Processing*. MIT Press, Cambridge, Massachusetts, pages 383–406.

Hofmann, Thomas and Jan Puzicha. 1998. "Statistical models for co-occurrence data." Memorandum, Massachussetts Institute of Technology Artificial Intelligence Laboratory, Cambridge, Massachusetts.

Jackendoff, Ray. 1972. *Semantic Interpretation in Generative Grammar*. MIT Press, Cambridge, Massachusetts.

Jelinek, Frederick and Robert L. Mercer. 1980. "Interpolated estimation of Markov source parameters from sparse data." In *Proceedings: Workshop on Pattern Recognition in Practice*, pages 381–397. Amsterdam. North Holland.

Johnson, Christopher R., Charles J. Fillmore, Esther J. Wood, Josef Ruppenhofer, Margaret Urban, Miriam R. L. Petruk, and Collin F. Baker. 2001. The FrameNet project: Tools for lexicon building. Version 0.7. Available at http://www.icsi.berkeley.edu/˜framenet/book.html.

Kipper, Karin, Hoa Trang Dang, William Schuler, and Martha Palmer. 2000. "Building a class-based verb lexicon using TAGs." In *TAG+5 Fifth International Workshop on Tree Adjoining Grammars and Related Formalisms*, Paris, May.

Lapata, Maria and Chris Brew. 1999. "Using subcategorization to resolve verb class ambiguity." In *Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora*, pages 266–274, College Park, Maryland.

Levin, Beth. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press, Chicago.

Levin, Beth and Malka Rappaport Hovav. 1996. "From lexical semantics to argument realization." Unpublished manuscript.

Marcus, Mitchell P., Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. "The Penn Treebank: Annotating predicate argument structure." In *ARPA Human Language Technology Workshop*, pages 114–119, Plainsboro, New Jersey. Morgan Kaufmann, San Francisco.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics* 19(2):313–330.

McCarthy, Diana. 2000. "Using semantic preferences to identify verbal participation in role switching alternations." In *Proceedings of the First Annual Meeting of the North American Chapter of the ACL (NAACL)*, pages 256–263, Seattle, Washington.

Miller, Scott, Heidi Fox, Lance Ramshaw, and Ralph Weischedel. 2000. "A novel use of statistical parsing to extract information from text." In *Proceedings of the First Annual Meeting of the North American Chapter of the ACL (NAACL)*, pages 226–233, Seattle, Washington.

Miller, Scott, David Stallard, Robert Bobrow, and Richard Schwartz. 1996. "A fully statistical approach to natural language interfaces." In *Proceedings of the 34th Annual Meeting of the ACL*, pages 55–61, Santa Cruz, California.

Misra, Vidya Niwas. 1966. *The Descriptive Technique of Panini*. Mouton, The Hague.

Pereira, Fernando, Naftali Tishby, and Lillian Lee. 1993. "Distributional clustering of English words." In *Proceedings of the 31st Annual Meeting of the ACL*, pages 183–190, Columbus, Ohio.

Pietra, Stephen Della, Vincent Della Pietra, and John Lafferty. 1997. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(4):380–393.

Pollard, Carl and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago.

Riloff, Ellen. 1993. Automatically constructing a dictionary for information extraction tasks. In *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI)*, pages 811–816, Washington, D.C.

Riloff, Ellen and Mark Schmelzenbach. 1998. "An empirical approach to conceptual case frame acquisition." In *Proceedings of the Sixth Workshop on Very Large Corpora*, pages 49–56, Montreal, Canada.

Rocher, Rosane. 1964. "Agent" et "Objet" chez Panini. *Journal of the American Oriental Society* 84:44–54.

Rooth, Mats. 1995. "Two-dimensional clusters in grammatical relations." In *AAAI Symposium on Representation and Acquisition of Lexical Knowledge*, Stanford, California.

Rooth, Mats, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. 1999. "Inducing a semantically annotated lexicon via EM-based clustering." In *Proceedings of the 37th Annual Meeting of the ACL*, pages 104–111, College Park, Maryland.

Schank, Roger C. 1972. Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology* 3:552–631.

Siegel, Sidney and N. John Castellan, Jr. 1988. *Nonparametric Statistics for the Behavioral Sciences*. 2nd ed. McGraw-Hill, New York.

Somers, Harold L. 1987. *Valency and Case in Computational Linguistics*. Edinburgh University Press, Edinburgh, Scotland.

Stallard, David. 2000. "Talk'n'travel: A conversational system for air travel planning." In *Proceedings of the Sixth*

Applied Natural Language Processing Conference (ANLP'00), pages 68–75.

Van Valin, Robert D. 1993. A synopsis of role and reference grammar. In Robert D. Van Valin, editor, *Advances in Role and Reference Grammar*. John Benjamins Publishing Company, Amsterdam, pages 1–166.

Winograd, Terry. 1972. Understanding natural language. *Cognitive Psychology*, 3(1). Reprinted as a book by Academic Press, 1972.