The Proposition Bank: An Annotated Corpus of Semantic Roles

Martha Palmer*
University of Pennsylvania

Daniel Gildea†
University of Rochester

Paul Kingsbury*
University of Pennsylvania

The Proposition Bank project takes a practical approach to semantic representation, adding a layer of predicate-argument information, or semantic role labels, to the syntactic structures of the Penn Treebank. The resulting resource can be thought of as shallow, in that it does not represent coreference, quantification, and many other higher-order phenomena, but also broad, in that it covers every instance of every verb in the corpus and allows representative statistics to be calculated.

We discuss the criteria used to define the sets of semantic roles used in the annotation process and to analyze the frequency of syntactic/semantic alternations in the corpus. We describe an automatic system for semantic role tagging trained on the corpus and discuss the effect on its performance of various types of information, including a comparison of full syntactic parsing with a flat representation and the contribution of the empty ''trace'' categories of the treebank.

1. Introduction

Robust syntactic parsers, made possible by new statistical techniques (Ratnaparkhi 1997; Collins 1999, 2000; Bangalore and Joshi 1999; Charniak 2000) and by the availability of large, hand-annotated training corpora (Marcus, Santorini, and Marcinkiewicz 1993; Abeillé 2003), have had a major impact on the field of natural language processing in recent years. However, the syntactic analyses produced by these parsers are a long way from representing the full meaning of the sentences that are parsed. As a simple example, in the sentences

(1) John broke the window.

(2) The window broke.

a syntactic analysis will represent *the window* as the verb's direct object in the first sentence and its subject in the second but does not indicate that it plays the same underlying **semantic role** in both cases. Note that both sentences are in the active voice

* Department of Computer and Information Science, University of Pennsylvania, 3330 Walnut Street, Philadelphia, PA 19104. Email: mpalmer@cis.upenn.edu.
† Department of Computer Science, University of Rochester, PO Box 270226, Rochester, NY 14627. Email: gildea@cs.rochester.edu.

Submission received: 9th December 2003; Accepted for publication: 11th July 2004

and that this alternation in subject between transitive and intransitive uses of the verb does not always occur; for example, in the sentences

(3) The sergeant played taps.

(4) The sergeant played.

the subject has the same semantic role in both uses. The same verb can also undergo syntactic alternation, as in

(5) Taps played quietly in the background.

and even in transitive uses, the role of the verb's direct object can differ:

(6) The sergeant played taps.

(7) The sergeant played a beat-up old bugle.

Alternation in the syntactic realization of semantic arguments is widespread, affecting most English verbs in some way, and the patterns exhibited by specific verbs vary widely (Levin 1993). The syntactic annotation of the Penn Treebank makes it possible to identify the subjects and objects of verbs in sentences such as the above examples. While the treebank provides semantic function tags such as *temporal* and *locative* for certain constituents (generally syntactic adjuncts), it does not distinguish the different roles played by a verb's grammatical subject or object in the above examples. Because the same verb used with the same syntactic subcategorization can assign different semantic roles, roles cannot be deterministically added to the treebank by an automatic conversion process with 100% accuracy. Our semantic-role annotation process begins with a rule-based automatic tagger, the output of which is then hand-corrected (see section 4 for details).

The Proposition Bank aims to provide a broad-coverage hand-annotated corpus of such phenomena, enabling the development of better domain-independent language understanding systems and the quantitative study of how and why these syntactic alternations take place. We define a set of underlying semantic roles for each verb and annotate each occurrence in the text of the original Penn Treebank. Each verb's roles are numbered, as in the following occurrences of the verb *offer* from our data:

(8) . . . [$_{\text{Arg0}}$ the company] to . . . *offer* [$_{\text{Arg1}}$ a 15% to 20% stake] [$_{\text{Arg2}}$ to the public] (wsj_0345)[1]

(9) . . . [$_{\text{Arg0}}$ Sotheby's] . . . *offered* [$_{\text{Arg2}}$ the Dorrance heirs] [$_{\text{Arg1}}$ a money-back guarantee] (wsj_1928)

(10) . . . [$_{\text{Arg1}}$ an amendment] *offered* [$_{\text{Arg0}}$ by Rep. Peter DeFazio] . . . (wsj_0107)

(11) . . . [$_{\text{Arg2}}$ Subcontractors] will be *offered* [$_{\text{Arg1}}$ a settlement] . . . (wsj_0187)

We believe that providing this level of semantic representation is important for applications including information extraction, question answering, and machine

1 Example sentences drawn from the treebank corpus are identified by the number of the file in which they occur. Constructed examples usually feature *John*.

translation. Over the past decade, most work in the field of information extraction has shifted from complex rule-based systems designed to handle a wide variety of semantic phenomena, including quantification, anaphora, aspect, and modality (e.g., Alshawi 1992), to more robust finite-state or statistical systems (Hobbs et al. 1997; Miller et al. 1998). These newer systems rely on a shallower level of semantic representation, similar to the level we adopt for the Proposition Bank, but have also tended to be very domain specific. The systems are trained and evaluated on corpora annotated for semantic relations pertaining to, for example, corporate acquisitions or terrorist events. The Proposition Bank (PropBank) takes a similar approach in that we annotate predicates' semantic roles, while steering clear of the issues involved in quantification and discourse-level structure. By annotating semantic roles for every verb in our corpus, we provide a more domain-independent resource, which we hope will lead to more robust and broad-coverage natural language understanding systems.

The Proposition Bank focuses on the argument structure of verbs and provides a complete corpus annotated with semantic roles, including roles traditionally viewed as arguments and as adjuncts. It allows us for the first time to determine the frequency of syntactic variations in practice, the problems they pose for natural language understanding, and the strategies to which they may be susceptible.

We begin the article by giving examples of the variation in the syntactic realization of semantic arguments and drawing connections to previous research into verb alternation behavior. In section 3 we describe our approach to semantic-role annotation, including the types of roles chosen and the guidelines for the annotators. Section 5 compares our PropBank methodology and choice of semantic-role labels to those of another semantic annotation project, FrameNet. We conclude the article with a discussion of several preliminary experiments we have performed using the PropBank annotations, and discuss the implications for natural language research.

2. Semantic Roles and Syntactic Alternation

Our work in examining verb alternation behavior is inspired by previous research into the linking between semantic roles and syntactic realization, in particular, the comprehensive study of Levin (1993). Levin argues that syntactic frames are a direct reflection of the underlying semantics; the sets of syntactic frames associated with a particular Levin class reflect underlying semantic components that constrain allowable arguments. On this principle, Levin defines verb **classes** based on the ability of particular verbs to occur or not occur in pairs of syntactic frames that are in some sense meaning-preserving (diathesis alternations). The classes also tend to share some semantic component. For example, the *break* examples above are related by a transitive/intransitive alternation called the causative/inchoative alternation. *Break* and other verbs such as *shatter* and *smash* are also characterized by their ability to appear in the middle construction, as in *Glass breaks/shatters/smashes easily. Cut*, a similar change-of-state verb, seems to share in this syntactic behavior and can also appear in the transitive (causative) as well as the middle construction: *John cut the bread, This loaf cuts easily*. However, it cannot also occur in the simple intransitive: *The window broke/*The bread cut*. In contrast, *cut* verbs can occur in the conative—*John valiantly cut/hacked at the frozen loaf, but his knife was too dull to make a dent in it*—whereas *break* verbs cannot: **John broke at the window.* The explanation given is that *cut* describes a series of actions directed at achieving the goal of separating some object into pieces. These actions consist of grasping an instrument with a sharp edge such as a knife and applying it in a cutting fashion to the object. It is possible for these actions to be

performed without the end result being achieved, but such that the *cutting* manner can still be recognized, for example, *John cut at the loaf*. Where *break* is concerned, the only thing specified is the resulting change of state, in which the object becomes separated into pieces.

VerbNet (Kipper, Dang, and Palmer 2000; Kipper, Palmer, and Rambow 2002) extends Levin's classes by adding an abstract representation of the syntactic frames for each class with explicit correspondences between syntactic positions and the semantic roles they express, as in *Agent REL Patient* or *Patient REL into pieces* for *break*.[2] (For other extensions of Levin, see also Dorr and Jones [2000] and Korhonen, Krymolowsky, and Marx [2003].) The original Levin classes constitute the first few levels in the hierarchy, with each class subsequently refined to account for further semantic and syntactic differences within a class. The argument list consists of thematic labels from a set of 20 such possible labels (Agent, Patient, Theme, Experiencer, etc.). The syntactic frames represent a mapping of the list of schematic labels to deep-syntactic arguments. Additional semantic information for the verbs is expressed as a set (i.e., conjunction) of semantic predicates, such as *motion, contact, transfer_info*. Currently, all Levin verb classes have been assigned thematic labels and syntactic frames, and over half the classes are completely described, including their semantic predicates. In many cases, the additional information that VerbNet provides for each class has caused it to subdivide, or use intersections of, Levin's original classes, adding an additional level to the hierarchy (Dang et al. 1998). We are also extending the coverage by adding new classes (Korhonen and Briscoe 2004).

Our objective with the Proposition Bank is not a theoretical account of how and why syntactic alternation takes place, but rather to provide a useful level of representation and a corpus of annotated data to enable empirical study of these issues. We have referred to Levin's classes wherever possible to ensure that verbs in the same classes are given consistent role labels. However, there is only a 50% overlap between verbs in VerbNet and those in the Penn TreeBank II, and PropBank itself does not define a set of classes, nor does it attempt to formalize the semantics of the roles it defines.

While lexical resources such as Levin's classes and VerbNet provide information about alternation patterns and their semantics, the frequency of these alternations and their effect on language understanding systems has never been carefully quantified. While learning syntactic subcategorization frames from corpora has been shown to be possible with reasonable accuracy (Manning 1993; Brent 1993; Briscoe and Carroll 1997), this work does not address the semantic roles associated with the syntactic arguments. More recent work has attempted to group verbs into classes based on alternations, usually taking Levin's classes as a gold standard (McCarthy 2000; Merlo and Stevenson 2001; Schulte im Walde 2000; Schulte im Walde and Brew 2002). But without an annotated corpus of semantic roles, this line of research has not been able to measure the frequency of alternations directly, or more generally, to ascertain how well the classes defined by Levin correspond to real-world data.

We believe that a shallow labeled dependency structure provides a feasible level of annotation which, coupled with minimal coreference links, could provide the foundation for a major advance in our ability to extract salient relationships from text. This will in turn improve the performance of basic parsing and generation

2 These can be thought of as a notational variant of tree-adjoining grammar elementary trees or tree-adjoining grammar partial derivations (Kipper, Dang, and Palmer 2000).

components, as well as facilitate advances in text understanding, machine translation, and fact retrieval.

3. Annotation Scheme: Choosing the Set of Semantic Roles

Because of the difficulty of defining a universal set of semantic or thematic roles covering all types of predicates, PropBank defines semantic roles on a verb-by-verb basis. An individual verb's semantic arguments are numbered, beginning with zero. For a particular verb, Arg0 is generally the argument exhibiting features of a Prototypical Agent (Dowty 1991), while Arg1 is a Prototypical Patient or Theme. No consistent generalizations can be made across verbs for the higher-numbered arguments, though an effort has been made to consistently define roles across members of VerbNet classes. In addition to verb-specific numbered roles, PropBank defines several more general roles that can apply to any verb. The remainder of this section describes in detail the criteria used in assigning both types of roles.

As examples of verb-specific numbered roles, we give entries for the verbs *accept* and *kick* below. These examples are taken from the guidelines presented to the annotators and are also available on the Web at http://www.cis.upenn.edu/~cotton/cgi-bin/pblex_fmt.cgi.

(12) Frameset **accept.01** "take willingly"

Arg0: Acceptor

Arg1: Thing accepted

Arg2: Accepted-from

Arg3: Attribute

Ex:[$_{\text{Arg0}}$ He] [$_{\text{ArgM-MOD}}$ would][$_{\text{ArgM-NEG}}$ n't] *accept* [$_{\text{Arg1}}$ anything of value] [$_{\text{Arg2}}$ from those he was writing about]. (wsj_0186)

(13) Frameset **kick.01** "drive or impel with the foot"

Arg0: Kicker

Arg1: Thing kicked

Arg2: Instrument (defaults to foot)

Ex1: [$_{\text{ArgM-DIS}}$ But] [$_{\text{Arg0}}$ two big New York banks$_i$] seem [$_{\text{Arg0}}$ *trace*$_i$] to have *kicked* [$_{\text{Arg1}}$ those chances] [$_{\text{ArgM-DIR}}$ away], [$_{\text{ArgM-TMP}}$ for the moment], [$_{\text{Arg2}}$ with the embarrassing failure of Citicorp and Chase Manhattan Corp. to deliver $7.2 billion in bank financing for a leveraged buy-out of United Airlines parent UAL Corp]. (wsj_1619)

Ex2: [$_{\text{Arg0}}$ John$_i$] tried [$_{\text{Arg0}}$ *trace*$_i$] to *kick* [$_{\text{Arg1}}$ the football], but Mary pulled it away at the last moment.

A set of roles corresponding to a distinct usage of a verb is called a **roleset** and can be associated with a set of syntactic frames indicating allowable syntactic variations in the expression of that set of roles. The roleset with its associated frames is called a

frameset. A polysemous verb may have more than one frameset when the differences in meaning are distinct enough to require a different set of roles, one for each frameset. The tagging guidelines include a "descriptor" field for each role, such as "kicker" or "instrument," which is intended for use during annotation and as documentation but does not have any theoretical standing. In addition, each frameset is complemented by a set of examples, which attempt to cover the range of syntactic alternations afforded by that usage. The collection of frameset entries for a verb is referred to as the verb's **frames file**.

The use of numbered arguments and their mnemonic names was instituted for a number of reasons. Foremost, the numbered arguments plot a middle course among many different theoretical viewpoints.[3] The numbered arguments can then be mapped easily and consistently onto any theory of argument structure, such as traditional theta role (Kipper, Palmer, and Rambow 2002), lexical-conceptual structure (Rambow et al. 2003), or Prague tectogrammatics (Hajičova and Kučerová 2002).

While most rolesets have two to four numbered roles, as many as six can appear, in particular for certain verbs of motion:[4]

(14) Frameset **edge.01** "move slightly"

Arg0: causer of motion	Arg3: start point
Arg1: thing in motion	Arg4: end point
Arg2: distance moved	Arg5: direction

Ex: [$_{\text{Arg0}}$ Revenue] *edged* [$_{\text{Arg5}}$ up] [$_{\text{Arg2-EXT}}$ 3.4%] [$_{\text{Arg4}}$ to $904 million] [$_{\text{Arg3}}$ from $874 million] [$_{\text{ArgM-TMP}}$ in last year's third quarter]. (wsj_1210)

Because of the use of Arg0 for agency, there arose a small set of verbs in which an external force could cause the Agent to execute the action in question. For example, in the sentence *. . . Mr. Dinkins would march his staff out of board meetings and into his private office . . .* (wsj_0765), the staff is unmistakably the marcher, the agentive role. Yet Mr. Dinkins also has some degree of agency, since he is causing the staff to do the marching. To capture this, a special tag, ArgA, is used for the agent of an induced action. This ArgA tag is used only for verbs of volitional motion such as *march* and *walk*, modern uses of *volunteer* (e.g., *Mary volunteered John to clean the garage,* or more likely the passive of that, *John was volunteered to clean the garage*), and, with some hesitation, *graduate* based on usages such as *Penn only graduates 35% of its students.* (This usage does not occur as such in the Penn Treebank corpus, although it is evoked in the sentence *No student should be permitted to be graduated from elementary school without having mastered the 3 R's at the level that prevailed 20 years ago.* (wsj_1286))

In addition to the semantic roles described in the rolesets, verbs can take any of a set of general, adjunct-like arguments (ArgMs), distinguished by one of the function tags shown in Table 1. Although they are not considered adjuncts, NEG for verb-level negation (e.g., *John did*n't *eat his peas*) and MOD for modal verbs (e.g., *John* would *eat*

3 By following the treebank, however, we are following a very loose government-binding framework.

4 We make no attempt to adhere to any linguistic distinction between arguments and adjuncts. While many linguists would consider any argument higher than Agr2 or Agr3 to be an adjunct, such arguments occur frequently enough with their respective verbs, or classes of verbs, that they are assigned a number in order to ensure consistent annotation.

Table 1
Subtypes of the ArgM modifier tag.

LOC: location	CAU: cause
EXT: extent	TMP: time
DIS: discourse connectives	PNC: purpose
ADV: general purpose	MNR: manner
NEG: negation marker	DIR: direction
MOD: modal verb	

everything else) are also included in this list to allow every constituent surrounding the verb to be annotated. DIS is also not an adjunct but is included to ease future discourse connective annotation.

3.1 Distinguishing Framesets

The criteria for distinguishing framesets are based on both semantics and syntax. Two verb meanings are distinguished as different framesets if they take different numbers of arguments. For example, the verb *decline* has two framesets:

(15) Frameset **decline.01** "go down incrementally"

Arg1: entity going down

Arg2: amount gone down by, EXT

Arg3: start point

Arg4: end point

Ex: . . . [$_{\text{Arg1}}$ its net income] *declining* [$_{\text{Arg2-EXT}}$ 42%] [$_{\text{Arg4}}$ to $121 million] [$_{\text{ArgM-TMP}}$ in the first 9 months of 1989]. (wsj_0067)

(16) Frameset **decline.02** "demure, reject"

Arg0: agent

Arg1: rejected thing

Ex: [$_{\text{Arg0}}$ A spokesman$_i$] *declined* [$_{\text{Arg1}}$ *trace*$_i$* to elaborate] (wsj_0038)

However, alternations which preserve verb meanings, such as causative/inchoative or object deletion, are considered to be one frameset only, as shown in the example (17). Both the transitive and intransitive uses of the verb *open* correspond to the same frameset, with some of the arguments left unspecified:

(17) Frameset **open.01** "cause to open"

Arg0: agent

Arg1: thing opened

Arg2: instrument

Ex1: [$_{\text{Arg0}}$ John] *opened* [$_{\text{Arg1}}$ the door]

Ex2: [$_{Arg1}$ The door] *opened*

Ex3: [$_{Arg0}$ John] *opened* [$_{Arg1}$ the door] [$_{Arg2}$ with his foot]

Moreover, differences in the syntactic type of the arguments do not constitute criteria for distinguishing among framesets. For example, *see.01* allows for either an NP object or a clause object:

(18) Frameset **see.01** "view"

Arg0: viewer

Arg1: thing viewed

Ex1: [$_{Arg0}$ John] *saw* [$_{Arg1}$ the President]

Ex2: [$_{Arg0}$ John] *saw* [$_{Arg1}$ the President collapse]

Furthermore, verb-particle constructions are treated as separate from the corresponding simplex verb, whether the meanings are approximately the same or not. Example (19-21) presents three of the framesets for *cut*:

(19) Frameset **cut.01** "slice"

Arg0: cutter

Arg1: thing cut

Arg2: medium, source

Arg3: instrument

Ex: [$_{Arg0}$ Longer production runs] [$_{ArgM-MOD}$ would] *cut* [$_{Arg1}$ inefficiencies from adjusting machinery between production cycles]. (wsj_0317)

(20) Frameset **cut.04** "cut off = slice"

Arg0: cutter

Arg1: thing cut (off)

Arg2: medium, source

Arg3: instrument

Ex: [$_{Arg0}$ The seed companies] *cut off* [$_{Arg1}$ the tassels of each plant]. (wsj_0209)

(21) Frameset **cut.05** "cut back = reduce"

Arg0: cutter

Arg1: thing reduced

Arg2: amount reduced by

Arg3: start point

Arg4: end point

Ex: "Whoa," thought John, "[$_{Arg0}$ I]$_i$'ve got [$_{Arg0}$ *trace*]$_i$ to start [$_{Arg0}$ *trace*]$_i$ *cutting back* [$_{Arg1}$ my intake of chocolate].

Note that the verb and particle do not need to be contiguous; (20) above could just as well be phrased *The seed companies cut the tassels of each plant off*.

For the WSJ text, there are frames for over 3,300 verbs, with a total of just over 4,500 framesets described, implying an average polysemy of 1.36. Of these verb frames, only 21.6% (721/3342) have more than one frameset, while less than 100 verbs have four or more. Each instance of a polysemous verb is marked as to which frameset it belongs to, with interannotator (ITA) agreement of 94%. The framesets can be viewed as extremely coarse-grained sense distinctions, with each frameset corresponding to one or more of the Senseval 2 WordNet 1.7 verb groupings. Each grouping in turn corresponds to several WordNet 1.7 senses (Palmer, Babko-Malaya, and Dang 2004).

3.2 Secondary Predications

There are two other functional tags which, unlike those listed above, can also be associated with numbered arguments in the frames files. The first one, EXT (extent), indicates that a constituent is a numerical argument on its verb, as in *climbed* 15% or *walked* 3 miles. The second, PRD (secondary predication), marks a more subtle relationship. If one thinks of the arguments of a verb as existing in a dependency tree, all arguments depend directly on the verb. Each argument is basically independent of the others. There are those verbs, however, which predict that there is a predicative relationship between their arguments. A canonical example of this is *call* in the sense of "attach a label to," as in *Mary called John an idiot*. In this case there is a relationship between *John* and *an idiot* (at least in Mary's mind). The PRD tag is associated with the Arg2 label in the frames file for this frameset, since it is predictable that the Arg2 predicates on the Arg1 *John*. This helps to disambiguate the crucial difference between the following two sentences:

predicative reading	*ditransitive reading*
Mary called John a doctor.	Mary called John a doctor.[5]
(LABEL)	(SUMMON)
Arg0: Mary	Arg0: Mary
Rel: called	Rel: called
Arg1: John (item being labeled)	Arg2: John (benefactive)
Arg2-PRD: a doctor (attribute)	Arg1: a doctor (thing summoned)

It is also possible for ArgMs to predicate on another argument. Since this must be decided on a case-by-case basis, the PRD function tag is added to the ArgM by the annotator, as in example (28).

5 This sense could also be stated in the dative: Mary called a doctor for John.

3.3 Subsumed Arguments

Because verbs which share a VerbNet class are rarely synonyms, their shared argument structure occasionally takes on odd characteristics. Of primary interest among these are the cases in which an argument predicted by one member of a class cannot be attested by another member of the same class. For a relatively simple example, consider the verb *hit*, in VerbNet classes 18.1 and 18.4. This takes three very obvious arguments:

(22) Frameset **hit** "strike"

Arg0: hitter

Arg1: thing hit, target

Arg2: instrument of hitting

Ex1: Agentive subject: "[$_{\text{Arg0}}$ He$_i$] digs in the sand instead of [$_{\text{Arg0}}$ *trace*$_i$] *hitting* [$_{\text{Arg1}}$ the ball], like a farmer," said Mr. Yoneyama. (wsj_1303)

Ex2: Instrumental subject: Dealers said [$_{\text{Arg1}}$ the shares] were *hit* [$_{\text{Arg2}}$ by fears of a slowdown in the U.S. economy]. (wsj_1015)

Ex3: All arguments: [$_{\text{Arg0}}$ John] *hit* [$_{\text{Arg1}}$ the tree] [$_{\text{Arg2}}$ with a stick].[6]

VerbNet classes 18.1 and 18.4 are filled with verbs of hitting, such as *beat*, *hammer*, *kick*, *knock*, *strike*, *tap*, and *whack*. For some of these the instrument of hitting is necessarily included in the semantics of the verb itself. For example, *kick* is essentially "hit with the foot" and *hammer* is exactly "hit with a hammer." For these verbs, then, the Arg2 might not be available, depending on how strongly the instrument is incorporated into the verb. *Kick*, for example, shows 28 instances in the treebank but only one instance of a (somewhat marginal) instrument:

(23) [$_{\text{ArgM-DIS}}$ But] [$_{\text{Arg0}}$ two big New York banks] seem to have *kicked* [$_{\text{Arg1}}$ those chances] [$_{\text{ArgM-DIR}}$away], [$_{\text{ArgM-TMP}}$ for the moment], [$_{\text{Arg2}}$ with the embarrassing failure of Citicorp and Chase Manhattan Corp. to deliver $7.2 billion in bank financing for a leveraged buy-out of United Airlines parent UAL Corp]. (wsj_1619)

Hammer shows several examples of Arg2s, but these are all metaphorical hammers:

(24) Despite the relatively strong economy, [$_{\text{Arg1}}$ junk bond prices$_i$] did nothing except go down, [$_{\text{Arg1}}$ *trace*$_i$] *hammered* [$_{\text{Arg2}}$ by a seemingly endless trail of bad news]. (wsj_2428)

Another perhaps more interesting case is that in which two arguments can be merged into one in certain syntactic situations. Consider the case of *meet*, which canonically takes two arguments:

(25) Frameset **meet** "come together"

Arg0: one party

6 The Wall Street Journal corpus contains no examples with both an agent and an instrument.

Arg1: the other party

Ex: [$_{\text{Arg0}}$ Argentine negotiator Carlos Carballo] [$_{\text{ArgM-MOD}}$ will] *meet* [$_{\text{Arg1}}$ with banks this week]. (wsj_0021)

It is perfectly possible, of course, to mention both meeting parties in the same constituent:

(26) [$_{\text{Arg0}}$ The economic and foreign ministers of 12 Asian and Pacific nations] [$_{\text{ArgM-MOD}}$ will] *meet* [$_{\text{ArgM-LOC}}$ in Australia] [$_{\text{ArgM-TMP}}$ next week] [$_{\text{ArgM-PRP}}$ to discuss global trade as well as regional matters such as transportation and telecommunications]. (wsj_0043)

In these cases there is an assumed or default Arg1 along the lines of "each other":

(27) [$_{\text{Arg0}}$ The economic and foreign ministers of 12 Asian and Pacific nations] [$_{\text{ArgM-MOD}}$ will] *meet* [$_{\text{Arg1-REC}}$ (with) each other] . . .

Similarly, verbs of attachment (*attach, tape, tie*, etc.) can express the things being attached as either one constituent or two:

(28) Frameset **connect.01** "attach"

Arg0: agent, entity causing two objects to be attached

Arg1: patient

Arg2: attached-to

Arg3: instrument

Ex1: The subsidiary also increased reserves by $140 million, however, and set aside an additional $25 million for [$_{\text{Arg1}}$ claims] *connected* [$_{\text{Arg2}}$ with Hurricane Hugo]. (wsj_1109)

Ex2: Machines using the 486 are expected to challenge higher-priced work stations and minicomputers in applications such as [$_{\text{Arg0}}$ so-called servers$_i$], [$_{\text{Arg0}}$ which$_i$] [$_{\text{Arg0}}$ *trace*$_i$] *connect* [$_{\text{Arg1}}$ groups of computers] [$_{\text{ArgM-PRD}}$ [together], and in computer-aided design. (wsj_0781)

3.4 Role Labels and Syntactic Trees

The Proposition Bank assigns semantic roles to nodes in the syntactic trees of the Penn Treebank. Annotators are presented with the roleset descriptions and the syntactic tree and mark the appropriate nodes in the tree with role labels. The lexical heads of constituents are not explicitly marked either in the treebank trees or in the semantic labeling layered on top of them. Annotators cannot change the syntactic parse, but they are not otherwise restricted in assigning the labels. In certain cases, more than one node may be assigned the same role. The annotation software does not require that the nodes being assigned labels be in any syntactic relation to the verb. We discuss the ways in which we handle the specifics of the treebank syntactic annotation style in this section.

3.4.1 Prepositional Phrases. The treatment of prepositional phrases is complicated by several factors. On one hand, if a given argument is defined as a "destination," then in a sentence such as *John poured the water into the bottle,* the destination of the water is clearly the bottle, not "into the bottle." The fact that the water is going into the bottle is inherent in the description "destination"; the preposition merely adds the specific information that the water will end up inside the bottle. Thus arguments should properly be associated with the NP heads of prepositional phrases. On the other hand, however, ArgMs which are prepositional phrases are annotated at the PP level, not the NP level. For the sake of consistency, then, numbered arguments are also tagged at the PP level. This also facilitates the treatment of multiword prepositions such as *out of*, *according to,* and *up to but not including*.[7]

(29) [$_{\text{Arg1}}$ Its net income] *declining* [$_{\text{Arg2-EXT}}$ 42%] [to $_{\text{Arg4}}$ $121 million] [$_{\text{ArgM-TMP}}$ in the first 9 months of 1989] (wsj_0067)

3.4.2 Traces and Control Verbs. The Penn Treebank contains empty categories known as traces, which are often coindexed with other constituents in the tree. When a trace is assigned a role label by an annotator, the coindexed constituent is automatically added to the annotation, as in

(30) [$_{\text{Arg0}}$ John$_i$] tried [$_{\text{Arg0}}$ *trace*$_i$] *to kick* [$_{\text{Arg1}}$ the football], but Mary pulled it away at the last moment.

Verbs such as *cause*, *force*, and *persuade*, known as object control verbs, pose a problem for the analysis and annotation of semantic structure. Consider a sentence such as *Commonwealth Edison said the ruling could force it to slash its 1989 earnings by $1.55 a share.* (wsj_0015). The Penn Treebank's analysis assigns a single sentential (S) constituent to the entire string *it to slash . . . a share*, making it a single syntactic argument to the verb *force*. In the PropBank annotation, we split the sentential complement into two semantic roles for the verb *force*, assigning roles to the noun phrase and verb phrase but not to the S node which subsumes them:

(31) Frameset **cause, force, persuade**, etc. "impelled action"

Arg0: agent

Arg1: impelled agent

Arg2: impelled action

Ex: Commonwealth Edison said [$_{\text{Arg0}}$ the ruling] [$_{\text{ArgM-MOD}}$ could] *force* [$_{\text{Arg1}}$ it] [$_{\text{Arg2-PRD}}$ to slash its 1989 earnings by $1.55 a share]. (wsj_0015)

In such a sentence, the object of the control verb will also be assigned a semantic role by the subordinate clause's verb:

(32) Commonwealth Edison said the ruling could force [$_{\text{Arg0}}$ it] to *slash* [$_{\text{Arg1}}$ its 1989 earnings] by [$_{\text{Arg2-by}}$ $1.55 a share]. (wsj_0015)

7 Note that *out of* is exactly parallel to *into*, but one is spelled with a space in the middle and the other isn't.

While *it* is the Arg0 of *force*, it is the Arg1 of *slash*. Similarly, subject control verbs such as *promise* result in the subject of the main clause being assigned two roles, one for each verb:

(33) [$_{\text{Arg0}}$ Mr. Bush's legislative package] *promises* [$_{\text{Arg2}}$ to cut emissions by 10 million tons—basically in half—by the year 2000]. (wsj_0146)

(34) [$_{\text{Arg0}}$ Mr. Bush's legislative package$_i$] promises [$_{\text{Arg0}}$ *trace*$_i$] to *cut* [$_{\text{Arg1}}$ emissions] [$_{\text{Arg2}}$ by 10 million tons—basically in half—] [$_{\text{ARGM-TMP}}$ by the year 2000].

We did not find a single case of a subject control verb used with a direct object and an infinitival clause (e.g., *John promised Mary to come*) in the Penn Treebank.

The cases above must be contrasted with verbs such as *expect*, often referred as exceptional case marking (ECM) verbs, where an infinitival subordinate clause is a single semantic argument:

(35) Frameset **expect** "look forward to, anticipate"

Arg0: expector

Arg1: anticipated event

Ex: Mr. Leinonen said [$_{\text{Arg0}}$ he] *expects* [$_{\text{Arg1}}$ Ford to meet the deadline easily]. (wsj_0064)

While *Ford* is given a semantic role for the verb *meet*, it is not given a role for *expect*.

3.4.3 Split Constituents. Most verbs of saying *(say, tell, ask, report*, etc.) have the property that the verb and its subject can be inserted almost anywhere within another of the verb's arguments. While the canonical realization is *John said (that) Mary was going to eat outside at lunchtime today*, it is common to say *Mary, John said, was going to eat outside at lunchtime today* or *Mary was going to eat outside, John said, at lunchtime today*. In this situation, there is no constituent holding the whole of the utterance while not also holding the verb of saying. We annotate these cases by allowing a single semantic role to point to the component pieces of the **split constituent** in order to cover the correct, discontinuous substring of the sentence.

(36) Frameset **say**

Arg0: speaker

Arg1: utterance

Arg2: listener

Ex: [$_{\text{Arg1}}$ By addressing those problems], [$_{\text{Arg0}}$ Mr. Maxwell] *said*, [$_{\text{Arg1}}$ the new funds have become "extremely attractive to Japanese and other investors outside the U.S."] (wsj_0029)

In the flat structure we have been using for example sentences, this looks like a case of repeated role labels. Internally, however, there is one role label pointing to multiple constituents of the tree, shown in Figure 1.

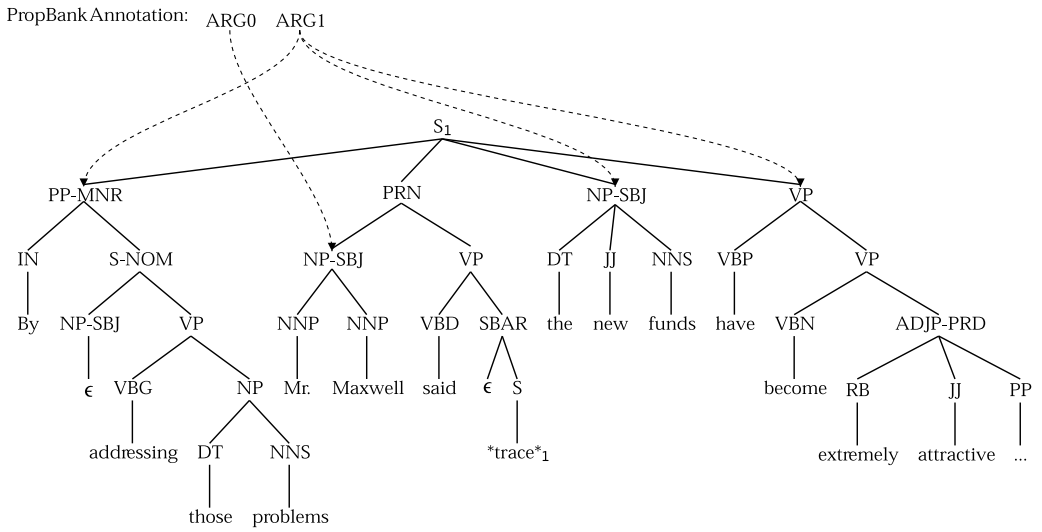

Figure 1
Split constituents: In this case, a single semantic role label points to multiple nodes in the original treebank tree.

4. The Propbank Development Process

Since the Proposition Bank consists of two portions, the lexicon of frames files and the annotated corpus, the process is similarly divided into framing and annotation.

4.1 Framing

The process of creating the frames files, that is, the collection of framesets for each lexeme, begins with the examination of a sample of the sentences from the corpus containing the verb under consideration. These instances are grouped into one or more major senses, and each major sense is turned into a single frameset. To show all the possible syntactic realizations of the frameset, many sentences from the corpus are included in the frames file, in the same format as the examples above. In many cases a particular realization will not be attested within the Penn Treebank corpus; in these cases, a constructed sentence is used, usually identified by the presence of the characters of John and Mary. Care was taken during the framing process to make synonymous verbs (mostly in the sense of "sharing a VerbNet Class") have the same framing, with the same number of roles and the same descriptors on those roles. Generally speaking, a given lexeme/sense pair required 10–15 minutes to frame, although highly polysemous verbs could require longer. With the 4,500+ framesets currently in place for PropBank, this is clearly a substantial time investment, and the frames files represent an important resource in their own right. We were able to use membership in a VerbNet class which already had consistent framing to project accurate frames files for up to 300 verbs. If the overlap between VerbNet and PropBank had been more than 50%, this number might have been higher.

4.2 Annotation

We begin the annotation process by running a rule-based argument tagger (Palmer, Rosenzweig, and Cotton 2001) on the corpus. This tagger incorporates an extensive lexicon, entirely separate from that used by PropBank, which encodes class-based

mappings between grammatical and semantic roles. The rule-based tagger achieved 83% accuracy on pilot data, with many of the errors due to differing assumptions made in defining the roles for a particular verb. The output of this tagger is then corrected by hand. Annotators are presented with an interface which gives them access to both the frameset descriptions and the full syntactic parse of any sentence from the treebank and allows them to select nodes in the parse tree for labeling as arguments of the predicate selected. For any verb they are able to examine both the descriptions of the arguments and the example tagged sentences, much as they have been presented here. The tagging is done on a verb-by-verb basis, known as lexical sampling, rather than all-words annotation of running text.

The downside of this approach is that it does not quickly provide a stretch of fully annotated text, needed for early assessment of the usefulness of the resource (see subsequent sections). For this reason a domain-specific subcorpus was automatically extracted from the entirety of the treebank, consisting of texts roughly primarily concerned with financial reporting and identified by the presence of a dollar sign anywhere in the text. This "financial" subcorpus comprised approximately one-third of the treebank and served as the initial focus of annotation.

The treebank as a whole contains 3,185 unique verb lemmas, while the financial subcorpus contains 1,826. These verbs are arrayed in a classic Zipfian distribution, with a few verbs occurring very often *(say*, for example, is the most common verb, with over 10,000 instances in its various inflectional forms) and most verbs occurring two or fewer times. As with the distribution of the lexical items themselves, the framesets also display a Zipfian distribution: A small number of verbs have many framesets (*go* has 20 when including phrasal variants, and *come, get, make, pass, take,* and *turn* each have more than a dozen) while the majority of verbs (2581/3342) have only one frameset. For polysemous verbs annotators had to determine which frameset was appropriate for a given usage in order to assign the correct argument structure, although this information was explicitly marked only during a separate pass.

Annotations were stored in a stand-off notation, referring to nodes within the Penn Treebank without actually replicating any of the lexical material or structure of that corpus. The process of annotation was a two-pass, blind procedure followed by an adjudication phase to resolve differences between the two initial passes. Both role labeling decisions and the choice of frameset were adjudicated.

The annotators themselves were drawn from a variety of backgrounds, from undergraduates to holders of doctorates, including linguists, computer scientists, and others. Undergraduates have the advantage of being inexpensive but tend to work for only a few months each, so they require frequent training. Linguists make the best overall judgments although several of our nonlinguist annotators also had excellent skills. The learning curve for the annotation task tended to be very steep, with most annotators becoming comfortable with the process within three days of work. This contrasts favorably with syntactic annotation, which has a much longer learning curve (Marcus, personal communication), and indicates one of the advantages of using a corpus already syntactically parsed as the basis of semantic annotation. Over 30 annotators contributed to the project, some for just a few weeks, some for up to three years. The framesets were created and annotation disagreements were adjudicated by a small team of highly trained linguists: Paul Kingsbury created the frames files and managed the annotators, and Olga Babko-Malaya checked the frames files for consistency and did the bulk of the adjudication.

We measured agreement between the two annotations before the adjudication step using the kappa statistic (Siegel and Castellan 1988), which is defined with respect to

the probability of interannotator agreement, $P(A)$, and the agreement expected by chance, $P(E)$:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

Measuring interannotator agreement for PropBank is complicated by the large number of possible annotations for each verb. For role identification, we expect agreement between annotators to be much higher than chance, because while any node in the parse tree can be annotated, the vast majority of arguments are chosen from the small number of nodes near the verb. In order to isolate the role classification decisions from this effect and avoid artifically inflating the kappa score, we split role identification (role vs. nonrole) from role classification (Arg0 vs. Arg1 vs. . . .) and calculate kappa for each decision separately. Thus, for the role identification kappa, the interannotator agreement probability $P(A)$ is the number of node observation agreements divided by the total number of nodes considered, which is the number of nodes in each parse tree multiplied by the number of predicates annotated in the sentence. All the PropBank data were annotated by two people, and in calculating kappa we compare these two annotations, ignoring the specific identities of the annotators for the predicate (in practice, agreement varied with the training and skill of individual annotators). For the role classification kappa, we consider only nodes that were marked as arguments by both annotators and compute kappa over the choices of possible argument labels. For both role identification and role classification, we compute kappa for two ways of treating ArgM labels. The first is to treat ArgM labels as arguments like any other, in which case ArgM-TMP, ArgM-LOC, and so on are considered separate labels for the role classification kappa. In the second scenario, we ignore ArgM labels, treating them as unlabeled nodes, and calculate agreement for identification and classification of numbered arguments only.

Kappa statistics for these various decisions are shown in Table 2. Agreement on role identification is very high (.99 under both treatments of ArgM), given the large number of obviously irrelevant nodes. Reassuringly, kappas for the more difficult role classification task are also high: .93 including all types of ArgM and .96 considering only numbered arguments. Kappas on the combined identification and classication decision, calculated over all nodes in the tree, are .91 including all subtypes of ArgM and .93 over numbered arguments only. Interannotator agreement among nodes that *either* annotator identified as an argument was .84, including ArgMs and .87, excluding ArgMs.

Discrepancies between annotators tended to be less on numbered arguments than on the selection of function tags, as shown in the confusion matrices of Tables 3 and 4.

Table 2
Interannotator agreement.

		$P(A)$	$P(E)$	κ
Including ArgM	Role identification	.99	.89	.93
	Role classification	.95	.27	.93
	Combined decision	.99	.88	.91
Excluding ArgM	Role identification	.99	.91	.94
	Role classification	.98	.41	.96
	Combined decision	.99	.91	.93

Table 3
Confusion matrix for argument labels, with ArgM labels collapsed into one category. Entries are a fraction of total annotations; true zeros are omitted, while other entries are rounded to zero.

	Arg0	Arg1	Arg2	Arg3	Arg4	ArgM
Arg0	0.288	0.006	0.001	0.000		0.000
Arg1		0.364	0.006	0.001	0.000	0.002
Arg2			0.074	0.001	0.001	0.003
Arg3				0.013	0.000	0.001
Arg4					0.011	0.000
ArgM						0.228

Certain types of functions, particularly those represented by the tags ADV, MNR, and DIS, can be difficult to distinguish. For example, in the sentence *Also, substantially lower Dutch corporate tax rates helped the company keep its tax outlay flat relative to earnings growth* (wsj_0132), the phrase *relative to earnings growth* could be interpreted as a manner adverbial (MNR), describing how the tax outlays were kept flat, or as a general-purpose adverbial (ADV), merely providing more information on the keeping event. Similarly, a word such as *then* can have several functions. It is canonically a temporal adverb marking time or a sequence of events (*. . . the Senate then broadened the list further . . .* (wsj_0101)) but can also mark a consequence of another action (*. . . if for any reason I don't have the values, then I won't recommend it*. (wsj_0331)) or simply serve as a placeholder in conversation (*It's possible then that Santa Fe's real estate . . . could one day fetch a king's ransom* (wsj_0331)). These three usages require three different taggings (TMP, ADV, and DIS, respectively) and can easily trip up an annotator.

The financial subcorpus was completely annotated and given a preadjudication release in June 2002. The fully annotated and adjudicated corpus was completed in March 2004. Both of these are available through the Linguistic Data Consortium, although because of the use of the stand-off notation, prior possession of the treebank is also necessary. The frames files are distributed separately and are available through the project Web site at http://www.cis.upenn.edu/~ace/.

Table 4
Confusion matrix among subtypes of ArgM, defined in Table 1. Entries are fraction of all ArgM labels. Entries are a fraction of all ArgM labels; true zeros are omitted, while other entries are rounded to zero.

	ADV	CAU	DIR	DIS	EXT	LOC	MNR	MOD	NEG	PNC	TMP
ADV	0.087	0.003	0.001	0.017	0.001	0.004	0.016	0.001	0.000	0.003	0.007
CAU		0.018	0.000	0.000		0.001	0.001			0.002	0.002
DIR			0.014		0.000	0.001	0.001				0.000
DIS				0.055	0.000	0.000	0.002	0.000	0.000	0.000	0.005
EXT					0.007	0.000	0.001			0.000	0.000
LOC						0.106	0.006	0.000	0.000	0.000	0.003
MNR							0.085	0.000	0.000	0.001	0.002
MOD								0.161	0.000		0.000
NEG									0.061		0.001
PNC										0.026	0.000
TMP											0.286

5. FrameNet and PropBank

The PropBank project and the FrameNet project at the International Computer Science Institute (Baker, Fillmore, and Lowe 1998) share the goal of documenting the syntactic realization of arguments of the predicates of the general English lexicon by annotating a corpus with semantic roles. Despite the two projects' similarities, their methodologies are quite different. FrameNet is focused on **semantic frames**,[8] which are defined as a schematic representation of situations involving various participants, props, and other conceptual roles (Fillmore 1976). The project methodology has proceeded on a **frame-by-frame basis**, that is, by first choosing a **semantic frame** (e.g., Commerce), defining the **frame** and its participants or **frame elements** (BUYER, GOODS, SELLER, MONEY), listing the various lexical predicates which invoke the **fram**e (*buy, sell,* etc.), and then finding example sentences of each predicate in a corpus (the British National Corpus was used) and annotating each **frame element** in each sentence. The example sentences were chosen primarily to ensure coverage of all the syntactic realizations of the **frame element**s, and simple examples of these realizations were preferred over those involving complex syntactic structure not immediately relevant to the lexical predicate itself. Only sentences in which the lexical predicate was used "in **frame**" were annotated. A word with multiple distinct senses would generally be analyzed as belonging to different **frames** in each sense but may only be found in the FrameNet corpus in the sense for which a **frame** has been defined. It is interesting to note that the **semantic frames** are a helpful way of generalizing between predicates; words in the same **frame** have been found frequently to share the same syntactic argument structure (Gildea and Jurafsky 2002). A more complete description of the FrameNet project can be found in Baker, Fillmore, and Lowe (1998) and Johnson et al. (2002), and the ramifications for automatic classification are discussed more thoroughly in Gildea and Jurafsky (2002).

In contrast with FrameNet, PropBank is aimed at providing data for training statistical systems and has to provide an annotation for every clause in the Penn Treebank, no matter how complex or unexpected. Similarly to FrameNet, PropBank also attempts to label semantically related verbs consistently, relying primarily on VerbNet classes for determining semantic relatedness. However, there is much less emphasis on the definition of the semantics of the class that the verbs are associated with, although for the relevant verbs additional semantic information is provided through the mapping to VerbNet. The PropBank semantic roles for a given VerbNet class may not correspond to the semantic elements highlighted by a particular FrameNet frame, as shown by the examples of Table 5. In this case, FrameNet's COMMERCE frame includes roles for Buyer (the receiver of the goods) and Seller (the receiver of the money) and assigns these roles consistently to two sentences describing the same event:

FrameNet annotation:

(37) $[_{\text{Buyer}}$ Chuck] *bought* $[_{\text{Goods}}$ a car] $[_{\text{Seller}}$ from Jerry] $[_{\text{Payment}}$ for \$1000].

(38) $[_{\text{Seller}}$ Jerry] *sold* $[_{\text{Goods}}$ a car] $[_{\text{Buyer}}$ to Chuck] $[_{\text{Payment}}$ for \$1000].

8 The authors apologize for the ambiguity between PropBank's "syntactic frames" and Framenet's "semantic frames." Syntactic frames refer to syntactic realizations. Semantic frames will appear herein in boldface.

Table 5
Comparison of frames.

PropBank		FrameNet
buy	*sell*	COMMERCE
Arg0: buyer	Arg0: seller	Buyer
Arg1: thing bought	Arg1: thing sold	Seller
Arg2: seller	Arg2: buyer	Payment
Arg3: price paid	Arg3: price paid	Goods
Arg4: benefactive	Arg4: benefactive	Rate/Unit

PropBank annotation:

(39) [$_{\text{Arg0}}$ Chuck] *bought* [$_{\text{Arg1}}$ a car] [$_{\text{Arg2}}$ from Jerry] [$_{\text{Arg3}}$ for $1000].

(40) [$_{\text{Arg0}}$ Jerry] *sold* [$_{\text{Arg1}}$ a car] [$_{\text{Arg2}}$ to Chuck] [$_{\text{Arg3}}$ for $1000].

PropBank requires an additional level of inference to determine who has possession of the car in both cases. However, FrameNet does not indicate that the subject in both sentences is an Agent, represented in PropBank by labeling both subjects as Arg0.[9] Note that the subject is not necessarily an agent, as in, for instance, the passive construction:

FrameNet annotation:

(41) [$_{\text{Goods}}$ A car] was *bought* [$_{\text{Buyer}}$ by Chuck].

(42) [$_{\text{Goods}}$ A car] was *sold* [$_{\text{Buyer}}$ to Chuck] [$_{\text{Seller}}$ by Jerry].

(43) [$_{\text{Buyer}}$ Chuck] was *sold* [$_{\text{Goods}}$ a car] [$_{\text{Seller}}$ by Jerry].

PropBank annotation:

(44) [$_{\text{Arg1}}$ A car] was *bought* [$_{\text{Arg0}}$ by Chuck].

(45) [$_{\text{Arg1}}$ A car] was *sold* [$_{\text{Arg2}}$ to Chuck] [$_{\text{Arg0}}$ by Jerry].

(46) [$_{\text{Arg2}}$ Chuck] was *sold* [$_{\text{Arg1}}$ a car] [$_{\text{Arg0}}$ by Jerry].

To date, PropBank has addressed only verbs, whereas FrameNet includes nouns and adjectives.[10] PropBank annotation also differs in that it takes place with reference to the Penn Treebank trees; not only are annotators shown the trees when analyzing a sentence, they are constrained to assign the semantic labels to portions of the sentence corresponding to nodes in the tree. Parse trees are not used in FrameNet; annotators mark the beginning and end points of **frame elements** in the text and add

9 FrameNet plans ultimately to represent agency is such examples using multiple inheritance of frames (Fillmore and Atkins 1998; Fillmore and Baker 2001).

10 New York University is currently in the process of annotating nominalizations in the Penn Treebank using the PropBank frames files and annotation interface, creating a resource to be known as NomBank.

a grammatical function tag expressing the **frame element**'s syntactic relation to the predicate.

6. A Quantitative Analysis of the Semantic-Role Labels

The stated aim of PropBank is the training of statistical systems. It also provides a rich resource for a distributional analysis of semantic features of language that have hitherto been somewhat inaccessible. We begin this section with an overview of general characteristics of the syntactic realization of the different semantic-role labels and then attempt to measure the frequency of syntactic alternations with respect to verb class membership. We base this analysis on previous work by Merlo and Stevenson (2001). In the following section we discuss the performance of a system trained to automatically assign the semantic-role labels.

6.1 Associating Role Labels with Specific Syntactic Constructions

We begin by simply counting the frequency of occurrence of roles in specific syntactic positions. In all the statistics given in this section, we do not consider past- or present-participle uses of the predicates, thus excluding any passive-voice sentences. The syntactic positions used are based on a few heuristic rules: Any NP under an S node in the treebank is considered a syntactic subject, and any NP under a VP is considered an object. In all other cases, we use the syntactic category of the argument's node in the treebank tree: for example, SBAR for sentential complements and PP for prepositional phrases. For prepositional phrases, as well as for noun phrases that are the object of a preposition, we include the preposition as part of our syntactic role: for example, PP-in, PP-with. Table 6 shows the most frequent semantic roles associated with various syntactic positions, while Table 7 shows the most frequent syntactic positions for various roles.

Tables 6 and 7 show overall statistics for the corpus, and some caution is needed in interpreting the results, as the semantic-role labels are defined on a per-frameset basis and do not necessarily have corpus-wide definitions. Nonetheless, a number of trends are apparent. Arg0, when present, is almost always a syntactic subject, while the subject is Arg0 only 79% of the time. This provides evidence for the notion of a thematic hierarchy in which the highest-ranking role present in a sentence is given the

Table 6
Most frequent semantic roles for each syntactic position.

Position	Total	Four most common roles (%)								Other roles (%)
Sub	37,364	Arg0	79.0	Arg1	16.8	Arg2	2.4	TMP	1.2	0.6
Obj	21,610	Arg1	84.0	Arg2	9.8	TMP	4.6	Arg3	0.8	0.8
S	10,110	Arg1	76.0	ADV	8.5	Arg2	7.5	PRP	2.4	5.5
NP	7,755	Arg2	34.3	Arg1	23.6	Arg4	18.9	Arg3	12.9	10.4
ADVP	5,920	TMP	30.3	MNR	22.2	DIS	19.8	ADV	10.3	17.4
MD	4,167	MOD	97.4	ArgM	2.3	Arg1	0.2	MNR	0.0	0.0
PP-in	3,134	LOC	46.6	TMP	35.3	MNR	4.6	DIS	3.4	10.1
SBAR	2,671	ADV	36.0	TMP	30.4	Arg1	16.8	PRP	7.6	9.2
RB	1,320	NEG	91.4	ArgM	3.3	DIS	1.6	DIR	1.4	2.3
PP-at	824	EXT	34.7	LOC	27.4	TMP	23.2	MNR	6.1	8.6

Table 7
Most frequent syntactic positions for each semantic role.

Roles	Total	Four most common syntactic positions (%)								Other positions (%)
Arg1	35,112	Obj	51.7	S	21.9	Subj	17.9	NP	5.2	3.4
Arg0	30,459	Subj	96.9	NP	2.4	S	0.2	Obj	0.2	0.2
Arg2	7,433	NP	35.7	Obj	28.6	Subj	12.1	S	10.2	13.4
TMP	6,846	ADVP	26.2	PP-in	16.2	Obj	14.6	SBAR	11.9	31.1
MOD	4,102	MD	98.9	ADVP	0.8	NN	0.1	RB	0.0	0.1
ADV	3,137	SBAR	30.6	S	27.4	ADVP	19.4	PP-in	3.1	19.5
LOC	2,469	PP-in	59.1	PP-on	10.0	PP-at	9.2	ADVP	6.4	15.4
MNR	2,429	ADVP	54.2	PP-by	9.6	PP-with	7.8	PP-in	5.9	22.5
Arg3	1,762	NP	56.7	Obj	9.7	Subj	8.9	ADJP	7.8	16.9
DIS	1,689	ADVP	69.3	CC	10.6	PP-in	6.2	PP-for	5.4	8.5

honor of subjecthood. Going from syntactic position to semantic role, the numbered arguments are more predictable than the non-predicate-specific adjunct roles. The two exceptions are the roles of "modal" (MOD) and "negative" (NEG), which as previously discussed are not syntactic adjuncts at all but were simply marked as ArgMs as the best means of tracking their important semantic contributions. They are almost always realized as auxiliary verbs and the single adverb (part-of-speech tag RB) *not*, respectively.

6.2 Associating Verb Classes with Specific Syntactic Constructions

Turning to the behavior of individual verbs in the PropBank data, it is interesting to see how much correspondence there is between verb classes proposed in the literature

Table 8
Semantic roles of verbs' subjects, for the verb classes of Merlo and Stevenson (2001).

Verb	Count	Relative frequency of semantic role				
		Arg0	Arg1	Arg2	ArgA	TMP
Unergative						
float	14	35.7	64.3			
hurry	2		100.0			
jump	125		97.6			2.4
leap	11		90.9			9.1
march	8	87.5			12.5	
race	4	75.0				25.0
rush	31	6.5	90.3			3.2
vault	1	100.0				
wander	3	100.0				
glide	1	100.0				
hop	34	97.1				2.9
jog	1	100.0				
scoot	1				100.0	
scurry	2	100.0				
skip	5	100.0				
tiptoe	2	100.0				

Table 8
(cont.)

Verb	Count	Relative frequency of semantic role				
		Arg0	Arg1	Arg2	ArgA	TMP
Unaccusative						
boil	1		100.0			
dissolve	4	75.0	25.0			
explode	7		100.0			
flood	5	80.0		20.0		
fracture	1	100.0				
melt	4	25.0	50.0			25.0
open	80	72.5	21.2	2.5		3.8
solidify	6	83.3	16.7			
collapse	36		94.4			5.6
cool	9	66.7	33.3			
widen	29	27.6	72.4			
change	148	65.5	33.8			0.7
clear	14	78.6	21.4			
divide	1	100.0				
simmer	5		100.0			
stabilize	33	45.5	54.5			
Object-Drop						
dance	2	100.0				
kick	5	80.0	20.0			
knit	1	100.0				
paint	4	100.0				
play	67	91.0	9.0			
reap	10	100.0				
wash	4	100.0				
yell	5	100.0				
borrow	36	100.0				
inherit	6	100.0				
organize	11	100.0				
sketch	1	100.0				
clean	4	100.0				
pack	7	100.0				
study	40	100.0				
swallow	5	80.0	20.0			
call	199	97.0	1.5	1.0		0.5

and the annotations in the corpus. Table 8 shows the PropBank semantic role labels for the subjects of each verb in each class. Merlo and Stevenson (2001) aim to automatically classify verbs into one of three categories: **unergative, unaccusative,** and **object-drop**. These three categories, more coarse-grained than the classes of Levin or VerbNet, are defined by the semantic roles they assign to a verb's subjects and objects in both transitive and intransitive sentences, as illustrated by the following examples:

Unergative: [$_{\text{Causal Agent}}$ The jockey] *raced* [$_{\text{Agent}}$ the horse] past the barn.

[$_{\text{Agent}}$ The horse] *raced* past the barn.

Unaccusative: [$_{\text{Causal Agent}}$ The cook] *melted* [$_{\text{Theme}}$ the butter] in the pan.

[$_{\text{Theme}}$ The butter] *melted* in the pan.

Object-Drop: [$_{\text{Agent}}$ The boy] *played* [$_{\text{Theme}}$ soccer].

[$_{\text{Agent}}$ The boy] *played*.

6.2.1 Predictions. In our data, the closest analogs to Merlo and Stevenson's three roles of Causal Agent, Agent, and Theme are ArgA, Arg0, and Arg1, respectively. We hypothesize that PropBank data will confirm

1. that the subject can take one of two roles (Arg0 or Arg1) for unaccusative and unergative verbs but only one role (Arg0) for object-drop verbs;
2. that Arg1s appear more frequently as subjects for intransitive unaccusatives than they do for intransitive unergatives.

In Table 8 we show counts for the semantic roles of the subjects of the Merlo and Stevenson verbs which appear in PropBank (80%), regardless of transitivity, in order to measure whether the data in fact reflect the alternations between syntactic and semantic roles that the verb classes predict. For each verb, we show counts only for occurrences tagged as belonging to the first frameset, reflecting the predominant or unmarked sense.

6.2.2 Results of Prediction 1. The object-drop verbs of Merlo and Stevenson do in fact show little variability in our corpus, with the subject almost always being Arg0. The unergative and unaccusative verbs show much more variability in the roles that can appear in the subject position, as predicted, although some individual verbs always have Arg0 as subject, presumably as a result of the small number of occurrences.

6.2.3 Results of Prediction 2. As predicted, there is in general a greater preponderance of Arg1 subjects for unaccusatives than for unergatives, with the striking exception of a few unergative verbs, such as *jump* and *rush*, whose subjects are almost always Arg1. *Jump* is being affected by the predominance of a financial-subcorpus sense used for stock reportage (79 out of 82 sentences), which takes *jump* as *rise dramatically: Jaguar shares jumped 23 before easing to close at 654, up 6.* (wsj_1957) *Rush* is being affected by a framing decision, currently being reconsidered, wherein *rush* was taken to mean *cause to move quickly*. Thus the *entity in motion* is tagged Arg1, as in *Congress* in *Congress would have rushed to pass a private relief bill*. (wsj_0946) The distinction between unergatives and unaccusatives is not apparent from the PropBank data in this table, since we are not distinguishing between transitives and intransitives, which is left for future experiments.

In most cases, the first frameset (numbered 1 in the PropBank frames files) is the most common, but in a few cases this is not the case because of the domain of the text. For example, the second frameset for *kick,* corresponding to the phrasal usage *kick in,* meaning *begin,* accounted for seven instances versus the five instances for frameset 1.

The phrasal frameset has a very different pattern, with the subject always corresponding to Arg1, as in

(47) [$_{\text{Arg1}}$ Several of those post-crash changes] *kicked in* [$_{\text{ArgM-TMP}}$ during Friday's one-hour collapse] and worked as expected, even though they didn't prevent a stunning plunge. (wsj_2417)

Statistics for all framesets of *kick* are shown in Table 9; the first row in Table 9 corresponds to the entry for *kick* in the "Object-Drop" section of Table 8.

Overall, these results support our hypotheses and also highlight the important role played by even the relatively coarse-grained sense tagging exemplified by the framesets.

7. Automatic Determination of Semantic-Role Labels

The stated goal of the PropBank is to provide training data for supervised automatic role labelers, and the project description cannot be considered complete without a discussion of PropBank's suitability for this purpose. One of PropBank's important features as a practical resource is that the sentences chosen for annotation are from the same Wall Street Journal corpus used for the original Penn Treebank project, and thus hand-checked syntactic parse trees are available for the entire data set. In this section, we examine the importance of syntactic information for semantic-role labeling by comparing the performance of a system based on gold-standard parses with one using automatically generated parser output. We then examine whether it is possible that the additional information contained in a full parse tree is negated by the errors present in automatic parser output, by testing a role-labeling system based on a flat or "chunked" representation of the input.

Gildea and Jurafsky (2002) describe a statistical system trained on the data from the FrameNet project to automatically assign semantic roles. The system first passed sentences through an automatic parser (Collins 1999), extracted syntactic features from the parses, and estimated probabilities for semantic roles from the syntactic and lexical features. Both training and test sentences were automatically parsed, as no hand-annotated parse trees were available for the corpus. While the errors introduced by the parser no doubt negatively affected the results obtained, there was no direct way of quantifying this effect. One of the systems evaluated for the Message Understanding Conference task (Miller et al. 1998) made use of an integrated syntactic and semantic model producing a full parse tree and achieved results comparable to other systems that did not make use of a complete parse. As in the FrameNet case, the parser was not

Table 9
Semantic roles for different frame sets of *kick*.

Frame set	Count	Relative frequency of semantic role				
		Arg0	Arg1	Arg2	ArgA	TMP
Unergative						
kick.01: drive or impel with the foot	5	80.0	20.0			
kick.02: *kick in*, begin	7		100.0			
kick.04: *kick off*, begin, inaugurate	3	100.0				

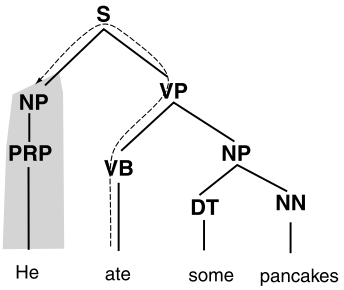

Figure 2
In this example, the **path** from the predicate *ate* to the argument NP *He* can be represented as VB↑VP↑S↓NP, with ↑ indicating upward movement in the parse tree and ↓ downward movement.

trained on the corpus for which semantic annotations were available, and the effect of better, or even perfect, parses could not be measured.

In our first set of experiments, the features and probability model of the Gildea and Jurafsky (2002) system were applied to the PropBank corpus. The existence of the hand-annotated treebank parses for the corpus allowed us to measure the improvement in performance offered by gold-standard parses.

7.1 System Description

Probabilities of a parse constituent belonging to a given semantic role are calculated from the following features:

The **phrase type** feature indicates the syntactic type of the phrase expressing the semantic roles: Examples include noun phrase (NP), verb phrase (VP), and clause (S).

The **parse tree path** feature is designed to capture the syntactic relation of a constituent to the predicate.[11] It is defined as the path from the predicate 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 2. Although the path is composed as a string of symbols, our systems treat the string as an atomic value. The path includes, as the first element of the string, the part of speech of the predicate, and as the last element, the phrase type or syntactic category of the sentence constituent marked as an argument.

The **position** feature simply indicates whether the constituent to be labeled occurs before or after the predicate. This feature is highly correlated with grammatical function, since subjects will generally appear before a verb and objects after. This feature may overcome the shortcomings of reading grammatical function from the parse tree, as well as errors in the parser output.

The **voice** feature distinguishes between active and passive verbs and is important in predicting semantic roles, because direct objects of active verbs correspond to subjects of passive verbs. An instance of a verb is considered passive if it is tagged as a past participle (e.g., *taken*), unless it occurs as a descendent verb phrase headed by any form of *have* (e.g., *has taken*) without an intervening verb phrase headed by any form of *be* (e.g., *has been taken*).

11 While the treebank has a "subject" marker on noun phrases, this is the only such grammatical function tag. The treebank does not explicitly represent which verb's subject the node is, and the subject tag is not typically present in automatic parser output.

The **headword** is a lexical feature and provides information about the semantic type of the role filler. Headwords of nodes in the parse tree are determined using the same deterministic set of headword rules used by Collins (1999).

The system attempts to predict argument roles in new data, looking for the highest-probability assignment of roles r_i to all constituents i in the sentence, given the set of features $F_i = \{pt_i, path_i, pos_i, v_i, h_i\}$ at each constituent in the parse tree, and the predicate p:

$$\text{argmax}_{r_{1\ldots n}} P(r_{1\ldots n}|F_{1\ldots n}, p)$$

We break the probability estimation into two parts, the first being the probability $P(r_i \mid F_i, p)$ of a constituent's role given our five features for the constituent and the predicate p. Because of the sparsity of the data, it is not possible to estimate this probability from the counts in the training data. Instead, probabilities are estimated from various subsets of the features and interpolated as a linear combination of the resulting distributions. The interpolation is performed over the most specific distributions for which data are available, which can be thought of as choosing the topmost distributions available from a back-off lattice, shown in Figure 3.

Next, the probabilities $P(r_i \mid F_i, p)$ are combined with the probabilities $P(\{r_{1\ldots n}\}|p)$ for a set of roles appearing in a sentence given a predicate, using the following formula:

$$P(r_{1\ldots n}|F_{1\ldots n}, p) \approx P(\{r_{1\ldots n}\}|p) \prod_i \frac{P(r_i|F_i, p)}{P(r_i|p)}$$

This approach, described in more detail in Gildea and Jurafsky (2002), allows interaction among the role assignments for individual constituents while making certain independence assumptions necessary for efficient probability estimation. In particular, we assume that sets of roles appear independent of their linear order and that the features F of a constituent are independent of other constituents' features given the constituent's role.

7.1.1 Results. We applied the same system, using the same features, to a preliminary release of the PropBank data. The data set used contained annotations for 72,109 predicate-argument structures containing 190,815 individual arguments and examples

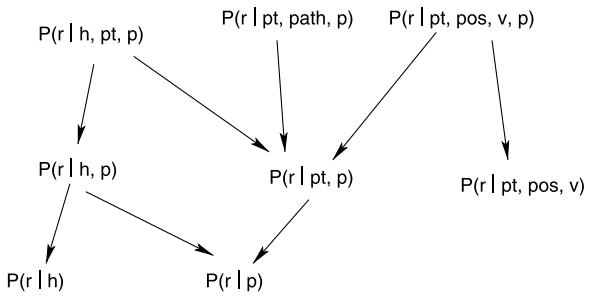

Figure 3
Back-off lattice with more specific distributions towards the top.

from 2,462 lexical predicates (types). In order to provide results comparable with the statistical parsing literature, annotations from section 23 of the treebank were used as the test set; all other sections were included in the training set. The preliminary version of the data used in these experiments was not tagged for WordNet word sense or PropBank frameset. Thus, the system neither predicts the frameset nor uses it as a feature.

The system was tested under two conditions, one in which it is given the constituents which are arguments to the predicate and merely has to predict the correct role, and one in which it has to both find the arguments in the sentence and label them correctly. Results are shown in Tables 10 and 11. Results for FrameNet are based on a test set of 8,167 individual labels from 4,000 predicate-argument structures. 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 (Gildea and Jurafsky 2002). For the PropBank data, with a test set of 8,625 individual labels, the threshold for significance is similar. There are 7,574 labels for which the predicate has been seen 10 or more times in training (third column of the tables).

Results for PropBank are similar to those for FrameNet, despite the smaller number of training examples for many of the predicates. The FrameNet data contained at least 10 examples from each predicate, while 12% of the PropBank data had fewer than 10 training examples. Removing these examples from the test set gives 82.8% accuracy with gold-standard parses and 80.9% accuracy with automatic parses.

7.1.2 Adding Traces. The gold-standard parses of the Penn Treebank include several types of information not typically produced by statistical parsers or included in their evaluation. Of particular importance are **traces**, empty syntactic categories which generally occupy the syntactic position in which another constituent could be interpreted and include a link to the relevant constituent. Traces are used to indicate cases of *wh*-extraction, antecedents of relative clauses, and control verbs exhibiting the syntactic phenomena of raising and "equi." Traces are intended to provide hints as to the semantics of individual clauses, and the results in Table 11 show that they do so effectively. When annotating syntactic trees, the PropBank annotators marked the traces along with their antecedents as arguments of the relevant verbs. In line 2 of Table 11, along with all our experiments with automatic parser output, traces were ignored, and the semantic-role label was assigned to the antecedent in both training and test data. In line 3 of Table 11, we assume that the system is given trace information, and in cases of trace chains, the semantic-role label is assigned to the trace in training and test conditions. Trace information boosts the performance of the system by roughly 5%. This indicates that systems capable of recovering traces (Johnson 2002; Dienes and Dubey 2003) could improve semantic-role labeling.

Table 10
Accuracy of semantic-role prediction (in percentages) for known boundaries (the system is given the constituents to classify).

	Accuracy		
	FrameNet	PropBank	PropBank > 10 examples
Automatic parses	82.0	79.9	80.9
Gold-standard parses		82.0	82.8

Table 11
Accuracy of semantic-role prediction (in percentages) for unknown boundaries (the system must identify the correct constituents as arguments and give them the correct roles).

	FrameNet		PropBank		PropBank > 10 examples	
	Precision	Recall	Precision	Recall	Precision	Recall
Automatic parses	64.6	61.2	68.6	57.8	69.9	61.1
Gold-standard parses			74.3	66.4	76.0	69.9
Gold-standard with traces			80.6	71.6	82.0	74.7

As our path feature is a somewhat unusual way of looking at parse trees, its behavior in the system warrants a closer look. The path feature is most useful as a way of finding arguments in the unknown boundary condition. Removing the path feature from the known-boundary system results in only a small degradation in performance, from 82.0% to 80.1%. One reason for the relatively small impact may be sparseness of the feature: 7% of paths in the test set are unseen in training data. The most common values of the feature are shown in Table 12, in which the first two rows correspond to standard subject and object positions. One reason for sparsity is seen in the third row: In the treebank, the adjunction of an adverbial phrase or modal verb can cause an additional VP node to appear in our path feature. We tried two variations of the path feature to address this problem. The first collapses sequences of nodes with the same label, for example, combining rows 2 and 3 of Table 12. The second variation uses only two values for the feature: NP under S (subject position) and NP under VP (object position). Neither variation improved performance in the known-boundary condition. As a gauge of how closely the PropBank semantic-role labels correspond to the path feature overall, we note that by always assigning the most common role for each path (for example, always assigning Arg0 to the subject position), and using no other features, we obtain the correct role 64.0% of the time, versus 82.0% for the complete system. Conditioning on the path and predicate, which allows the subject of different verbs to receive different labels but does not allow for alternation behavior within a verb's argument structure, yields an accuracy rate of 76.6%.

Table 13 shows the performance of the system broken down by the argument types in the gold standard. Results are shown for the unknown-boundaries condition, using gold-standard parses and traces (last row, middle two columns of Table 11). The

Table 12
Common values (in percentages) for parse tree path in PropBank data, using gold-standard parses.

Path	Frequency
VB↑VP↓NP	17.6
VB↑VP↑S↓NP	16.4
VB↑VP↑VP↑S↓NP	7.8
VB↑VP↓PP	7.6
VB↑VP↓PP↓NP	7.3
VB↑VP↓SBAR↓S	4.3
VB↑VP↓S	4.3
VB↑VP↓ADVP	2.4
Others (n = 1,031)	76.0

Table 13
Accuracy of semantic-role prediction for unknown boundaries (the system must identify the correct constituents as arguments and give them the correct roles).

Role	Number	Precision	Labeled recall	Unlabeled recall
Arg0	1,197	94.2%	88.9%	92.2%
Arg1	1,436	95.4	82.5	88.9
Arg2	229	79.0	64.2	77.7
Arg3	61	71.4	49.2	54.1
Arg4	31	91.7	71.0	83.9
ArgM	127	59.6	26.8	52.0
ArgM-ADV	85	59.1	30.6	55.3
ArgM-DIR	49	76.7	46.9	61.2
ArgM-DIS	65	40.0	18.5	55.4
ArgM-EXT	18	81.2	72.2	77.8
ArgM-LOC	95	60.7	38.9	62.1
ArgM-MNR	80	62.7	40.0	63.8
ArgM-MOD	95	77.6	40.0	43.2
ArgM-NEG	40	63.6	17.5	40.0
ArgM-PRD	3	0.0	0.0	33.3
ArgM-PRP	54	70.0	25.9	37.0
ArgM-TMP	325	72.4	45.2	64.6

"Labeled Recall" column shows how often the semantic-role label is correctly identified, while the "Unlabeled recall" column shows how often a constituent with the given role is correctly identified as being a semantic role, even if it is labeled with the wrong role. The more central, numbered roles are consistently easier to identify than the adjunct-like ArgM roles, even when the ArgM roles have preexisting Treebank function tags.

7.2 The Relation of Syntactic Parsing and Semantic-Role Labeling

Many recent information extraction systems for limited domains have relied on finite-state systems that do not build a full parse tree for the sentence being analyzed. Among such systems, Hobbs et al. (1997) built finite-state recognizers for various entities, which were then cascaded to form recognizers for higher-level relations, while Ray and Craven (2001) used low-level "chunks" from a general-purpose syntactic analyzer as observations in a trained hidden Markov model. Such an approach has a large advantage in speed, as the extensive search of modern statistical parsers is avoided. It is also possible that this approach may be more robust to error than parsers. Our experiments working with a flat, "chunked" representation of the input sentence, described in more detail in Gildea and Palmer (2002), test this finite-state hypothesis. In the chunked representation, base-level constituent boundaries and labels are present, but there are no dependencies between constituents, as shown by the following sample sentence:

(48) $[_{NP}$ Big investment banks] $[_{VP}$ refused to step] $[_{ADVP}$ up] $[_{PP}$ to]
$[_{NP}$ the plate] $[_{VP}$ to support] $[_{NP}$ the beleaguered floor traders] $[_{PP}$ by]
$[_{VP}$ buying] $[_{NP}$ bigblocks] $[_{PP}$ of] $[_{NP}$ stock], $[_{NP}$ traders] $[_{VP}$ say]. (wsj_2300)

Our chunks were derived from the treebank trees using the conversion described by Tjong Kim Sang and Buchholz (2000). Thus, the experiments were carried out using

gold-standard rather than automatically derived chunk boundaries, which we believe will provide an upper bound on the performance of a chunk-based system. Distance in chunks from the predicate was used in place of the parser-based path feature.

The results in Table 14 show that full parse trees are much more effective than the chunked representation for labeling semantic roles. This is the case even if we relax the scoring criteria to count as correct all cases in which the system correctly identifies the first chunk belonging to an argument (last row of Table 14).

As an example for comparing the behavior of the tree-based and chunk-based systems, consider the following sentence, with human annotations showing the arguments of the predicate *support*:

(49) [$_{\text{Arg0}}$ Big investment banks] refused to step up to the plate to *support* [$_{\text{Arg1}}$ the beleaguered floor traders] [$_{\text{MNR}}$ by buying big blocks of stock], traders say.

Our system based on automatic parser output assigned the following analysis:

(50) Big investment banks refused to step up to the plate to *support* [$_{\text{Arg1}}$ the beleaguered floor traders] [$_{\text{MNR}}$ by buying big blocks of stock], traders say.

In this case, the system failed to find the predicate's Arg0 relation, because it is syntactically distant from the verb *support*. The original treebank syntactic tree contains a trace which would allow one to recover this relation, coindexing the empty subject position of *support* with the noun phrase *Big investment banks*. However, our automatic parser output does not include such traces. The system based on gold-standard trees and incorporating trace information produced exactly the correct labels:

(51) [$_{\text{Arg0}}$ Big investment banks] refused to step up to the plate to *support* [$_{\text{Arg1}}$ the beleaguered floor traders] [$_{\text{MNR}}$ by buying big blocks of stock], traders say.

The system based on (gold-standard) chunks assigned the following semantic-role labels:

(52) Big investment banks refused to step up to [$_{\text{Arg0}}$ the plate] to *support* [$_{\text{Arg1}}$ the beleaguered floor traders] by buying big blocks of stock, traders say.

Here, as before, the true Arg0 relation is not found, and it would be difficult to imagine identifying it without building a complete syntactic parse of the sentence. But now,

Table 14
Summary of results for unknown-boundary condition.

	Precision	Recall
Gold parse	74.3%	66.4%
Auto parse	68.6	57.8
Chunk	27.6	22.0
Chunk, relaxed scoring	49.5	35.1

unlike in the tree-based output, the Arg0 label is mistakenly attached to a noun phrase immediately before the predicate. The Arg1 relation in direct-object position is fairly easily identifiable in the chunked representation as a noun phrase directly following the verb. The prepositional phrase expressing the Manner relation, however, is not identified by the chunk-based system. The tree-based system's path feature for this constituent is VB↑VP↓PP, which identifies the prepositional phrase as attaching to the verb and increases its probability of being assigned an argument label. The chunk-based system sees this as a prepositional phrase appearing as the second chunk after the predicate. Although this may be a typical position for the Manner relation, the fact that the preposition attaches to the predicate rather than to its direct object is not represented.

Participants in the 2004 CoNLL semantic-labeling shared task (Carreras and Màrquez 2004) have reported higher results for chunk-based systems, but to date chunk-based systems have not closed the gap with the state-of-the-art results based on parser output.

7.2.1 Parsing and Models of Syntax. While treebank parsers such as that of Collins (1999) return much richer representations than a chunker, they do not include a great deal of the information present in the original Penn Treebank. Specifically, long-distance dependencies indicated by traces in the treebank are crucial for semantic interpretation but do not affect the constituent recall and precision metrics most often used to evaluate parsers and are not included in the output of the standard parsers.

Gildea and Hockenmaier (2003) present a system for labeling PropBank's semantic roles based on a statistical parser for combinatory categorial grammar (CCG) (Steedman 2000). The parser, described in detail in Hockenmaier and Steedman (2002), is trained on a version of the Penn Treebank automatically converted to CCG representations. The conversion process uses the treebank's trace information to make underlying syntactic relations explicit. For example, the same CCG-level relation appears between a verb and its direct object whether the verb is used in a simple transitive clause, a relative clause, or a question with *wh*-extraction. Using the CCG-based parser, Gildea and Hockenmaier (2003) find a 2% absolute improvement over the Collins parser in identifying core or numbered PropBank arguments. This points to the shortcomings of evaluating parsers purely on constituent precision and recall; we feel that a dependency-based evaluation (e.g., Carroll, Briscoe, and Sanfilippo 1998) is more relevant to real-world applications.

8. Conclusion

The Proposition Bank takes the comprehensive corpus annotation of the Penn Treebank one step closer to a detailed semantic representation by adding semantic-role labels. On analyzing the data, the relationships between syntax and semantic structures are more complex than one might at first expect. Alternations in the realization of semantic arguments of the type described by Levin (1993) turn out to be common in practice as well as in theory, even in the limited genre of *Wall Street Journal* articles. Even so, by using detailed guidelines for the annotation of each individual verb, rapid consistent annotation has been achieved, and the corpus is available through the Linguistic Data Consortium. For information on obtaining the frames file, please consult http://www.cis.upenn.edu/~ace/.

The broad-coverage annotation has proven to be suitable for training automatic taggers, and in addition to ourselves there is a growing body of researchers engaged in this task. Chen and Rambow (2003) make use of extracted tree-adjoining grammars. Most recently, the Gildea and Palmer (2002) scores presented here have been improved markedly through the use of support-vector machines as well as additional features for named entity tags, headword POS tags, and verb clusters for back-off (Pradhan et al. 2003) and using maximum-entropy classifiers (He and Gildea 2004, Xue and Palmer 2004). This group also used Charniak's parser instead of Collins's and tested the system on TDT data. The performance on a new genre is lower, as would be expected.

Despite the complex relationship between syntactic and semantic structures, we find that statistical parsers, although computationally expensive, do a good job of providing information relevant for this level of semantic interpretation. In addition to the constituent structure, the headword information, produced as a side product, is an important feature. Automatic parsers, however, still have a long way to go. Our results using hand-annotated parse trees including traces show that improvements in parsing should translate directly into more accurate semantic representations.

There has already been a demonstration that a preliminary version of these data can be used to simplify the effort involved in developing information extraction (IE) systems. Researchers were able to construct a reasonable IE system by simply mapping specific Arg labels for a set of verbs to template slots, completely avoiding the necessity of building explicit regular expression pattern matchers (Surdeanu et al. 2003). There is equal hope for advantages for machine translation, and proposition banks in Chinese (Xue and Palmer 2003) and Korean are already being built, focusing where possible on parallel data. The general approach ports well to new languages, with the major effort continuing to go into the creation of frames files for verbs.

There are many directions for future work. Our preliminary linguistic analyses have merely scratched the surface of what is possible with the current annotation, and yet it is only a first approximation at capturing the richness of semantic representation. Annotation of nominalizations and other noun predicates is currently being added by New York University, and a Phase II (Babko-Malaya et al.) that will include eventuality variables, nominal references, additional sense tagging, and discourse connectives is underway.

We have several plans for improving the performance of our automatic semantic-role labeling. As a first step we are producing a version of PropBank that uses more informative thematic labels based on VerbNet thematic labels (Kipper, Palmer, and Rambow 2002). We are also working with FrameNet to produce a mapping between our annotation and theirs which will allow us to merge the two annotated data sets. Finally, we will explore alternative machine-learning approaches and closer integration of semantic-role labeling and sense tagging with the parsing process.

Acknowledgments

This work was funded by DOD grant MDA904-00C-2136, NSF grant IIS-9800658, and the Institute for Research in Cognitive Science at the University of Pennsylvania NSF-STC grant SBR-89-20230. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We are indebted to Scott Cotton, our staff programmer, for his untiring efforts and his lovely annotation tool, to Joseph Rosenzweig for the initial automatic semantic-role labeling of the Penn Treebank, and to Ben Snyder for programming support and computing agreement figures. We would also like to thank Mitch Marcus, Aravind Joshi, Olga Babko-Malaya, Hoa Trang Dang, Christiane Fellbaum, and Betsy Klipple for their extremely useful and insightful guidance and our many hard-working

annotators, especially Kate Forbes, Ilana Streit, Ann Delilkin, Brian Hertler, Neville Ryant, and Jill Flegg, for all of their help.

References

Abeillé, Anne, editor. 2003. *Building and Using Parsed Corpora*. Language and Speech series. Kluwer, Dordrecht.

Alshawi, Hiyan, editor. 1992. *The Core Language Engine*. MIT Press, Cambridge, MA.

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

Bangalore, Srinivas and Aravind K. Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–265.

Brent, Michael R. 1993. From grammar to lexicon: Unsupervised learning of lexical syntax. *Computational Linguistics*, 19(2): 243–262.

Briscoe, Ted and John Carroll. 1997. Automatic extraction of subcategorization from corpora. In *Fifth Conference on Applied Natural Language Processing*, pages 356–363, Washington, DC. ACL.

Carreras, Xavier and Lluís Màrquez. 2004. Introduction to the CoNLL-2004 shared task: Semantic role labeling. In *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*, pages 89–97, Boston.

Carroll, John, Ted Briscoe, and Antonio Sanfilippo. 1998. Parser evaluation: A survey and a new proposal. In *Proceedings of the First International Conference on Language Resources and Evaluation*, pages 447–454, Granada, Spain.

Charniak, Eugene. 2000. A maximum-entropy-inspired parser. In *Proceedings of the First Annual Meeting of the North American Chapter of the ACL (NAACL)*, pages 132–139, Seattle.

Chen, John and Owen Rambow. 2003. Use of deep linguistic features for the recognition and labeling of semantic arguments. In Michael Collins and Mark Steedman, editors, *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, Sapporo, Japan, pages 41–48.

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

Collins, Michael. 2000. Discriminative reranking for natural language parsing. In *Proceedings of the International Conference on Machine Learning (ICML)*, Stanford, CA.

Dang, Hoa Trang, Karin Kipper, Martha Palmer, and Joseph Rosenzweig. 1998. Investigating regular sense extensions based on intersective Levin classes. In *COLING/ACL-98*, pages 293–299, Montreal. ACL.

Dienes, Peter and Amit Dubey. 2003. Antecedent recovery: Experiments with a trace tagger. In *2003 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Sapporo, Japan.

Dorr, Bonnie J. and Douglas Jones. 2000. Acquisition of semantic lexicons: Using word sense disambiguation to improve precision. In Evelyn Viegas, editor, *Breadth and Depth of Semantic Lexicons*. Kluwer Academic, Norwell, MA, pages 79–98.

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

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.

Fillmore, Charles J. and B. T. S. Atkins. 1998. FrameNet and lexicographic relevance. In *Proceedings of the First International Conference on Language Resources and Evaluation*, Granada, Spain.

Fillmore, Charles J. and Collin F. Baker. 2001. Frame semantics for text understanding. In *Proceedings of NAACL WordNet and Other Lexical Resources Workshop*, Pittsburgh, June.

Gildea, Daniel and Julia Hockenmaier. 2003. Identifying semantic roles using combinatory categorial grammar. In *2003 Conference on Empirical Methods in Natural Language Processing (EMNLP*), Sapporo, Japan.

Gildea, Daniel and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics,* 28(3):245–288.

Gildea, Daniel and Martha Palmer. 2002. The necessity of syntactic parsing for predicate argument recognition. In *Proceedings of the 40th Annual Conference of the Association for Computational Linguistics (ACL-02),* pages 239–246, Philadelphia.

Hajičová, Eva and Ivona Kučerová. 2002. Argument/valency structure in PropBank, LCS Database and Prague Dependency Treebank: A comparative pilot study. In

Proceedings of the Third International Conference on Language Resources and Evaluation (LREC 2002), Las Palmas, Spain, pages 846–851. ELRA.

He, Shan and Daniel Gildea. 2004. Semantic roles labeling by maximum entropy model. Technical Report 847, University of Rochester.

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, MA, pages 383–406.

Hockenmaier, Julia and Mark Steedman. 2002. Generative models for statistical parsing with combinatory categorial grammar. In *Proceedings of the 40th Annual Conference of the Association for Computational Linguistics (ACL-02)*, pages 335–342, Philadelphia.

Johnson, Christopher R., Charles J. Fillmore, Miriam R. L. Petruck, Collin F. Baker, Michael Ellsworth, Josef Ruppenhofer, and Esther J. Wood. 2002. FrameNet: Theory and practice. Version 1.0, available at http://www.icsi.berkeley.edu/framenet/.

Johnson, Mark. 2002. A simple pattern-matching algorithm for recovering empty nodes and their antecedents. In *Proceedings of the 40th Annual Conference of the Association for Computational Linguistics (ACL-02),* Philadelphia.

Kipper, Karin, Hoa Trang Dang, and Martha Palmer. 2000. Class-based construction of a verb lexicon. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-2000)*, Austin, TX, July–August.

Kipper, Karin, Martha Palmer, and Owen Rambow. 2002. Extending PropBank with VerbNet semantic predicates. Paper presented at Workshop on Applied Interlinguas, AMTA-2002, Tiburon, California, October.

Korhonen, Anna and Ted Briscoe. 2004. Extended lexical-semantic classification of English verbs. In *Proceedings of the HLT/NAACL Workshop on Computational Lexical Semantics,* Boston.

Korhonen, Anna, Yuval Krymolowsky, and Zvika Marx. 2003. Clustering polysemic subcategorization frame distributions semantically. In *Proceedings of the 41st Annual Conference of the Association for Computational Linguistics (ACL-03),* Sapporo, Japan.

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

Manning, Christopher D. 1993. Automatic acquisition of a large subcategorization dictionary from corpora. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics,* pages 235–242, Ohio State University, Columbus.

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 1st Annual Meeting of the North American Chapter of the ACL (NAACL)*, pages 256–263, Seattle.

Merlo, Paola and Suzanne Stevenson. 2001. Automatic verb classification based on statistical distribution of argument structure. *Computational Linguistics,* 27(3):373–408.

Miller, Scott, Michael Crystal, Heidi Fox, Lance Ramshaw, Richard Schwartz, Rebecca Stone, Ralph Weischedel, and the Annotation Group. 1998. Algorithms that learn to extract information—BBN: Description of the SIFT system as used for MUC-7. In *Proceedings of the Seventh Message Understanding Conference (MUC-7),* April.

Babko-Malaya, Olga, Martha Palmer, Nianwen Xue, Aravind Joshi, Seth Kulick, Proposition Bank II: Delving deeper, frontiers in corpus annotation, Workshop in conjunction with HLT/NAACL 2004, Boston, MA, May 6, 2004.

Palmer, Martha, Olga Babko-Malaya, and Hoa Trang Dang. 2004. Different sense granularities for different applications. In *Second Workshop on Scalable Natural Language Understanding Systems at HLT/NAACL-04*, Boston.

Palmer, Martha, Joseph Rosenzweig, and Scott Cotton. 2001. Predicate argument analysis of the Penn Treebank. In *Proceedings of HLT 2001, First International Conference on Human Language Technology Research,* San Diego, CA, March.

Pradhan, S., K. Hacioglu, W. Ward, J. Martin, and Daniel Jurafsky. 2003. Semantic role parsing: Adding semantic structure to unstructured text. In *Proceedings of the International Conference on Data Mining (ICDM-2003),* Melbourne, FL.

Rambow, Owen, Bonnie J. Dorr, Karin Kipper, Ivona Kučerová, and Martha Palmer. 2003. Automatically deriving tectogrammatical labels from other resources: A comparison of semantic labels across frameworks. *Prague Bulletin of Mathematical Linguistics*, vol. 79–80, pages 23–35.

Ratnaparkhi, Adwait. 1997. A linear observed time statistical parser based on maximum entropy models. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing,* pages 1–10, Providence, ACL.

Ray, Soumya and Mark Craven. 2001. Representing sentence structure in hidden Markov model for information extraction. In *Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01),* Seattle.

Schulte im Walde, Sabine. 2000. Clustering verbs semantically according to their alternation behaviour. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING-00),* pages 747–753, Saarbrücken, Germany.

Schulte im Walde, Sabine and Chris Brew. 2002. Inducing German semantic verb classes from purely syntactic subcategorisation information. In *Proceedings of the 40th Annual Conference of the Association for Computational Linguistics (ACL-02)*, pages 223–230, Philadelphia.

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

Steedman, Mark. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA.

Surdeanu, Mihai, Sanda Harabagiu, John Williams, and Paul Aarseth. 2003. Using predicate-argument structures for information extraction. In *Proceedings of the 41st Annual Conference of the Association for Computational Linguistics (ACL-03),* Sapporo, Japan, pages 8–15.

Tjong Kim Sang, Erik F. and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal.

Xue, Nianwen, and Martha Palmer. 2004. Calibrating Features for Semantic Role Labeling, Empirical Methods in Natural Language Processing Conference, in conjunction with the *42nd Meeting of the Association for Computational Linguistics (ACL-04)*, Barcelona, Spain, July 21–26.

Xue, Nianwen, and Martha Palmer. 2004. Annotating the Propositions in the Penn Chinese Treebank Second SIGHAN Workshop on Chinese Language Processing, held in conjunction with ACL-03, Sapporo, Japan, pages 47–54, July.