Linguistic Annotation

Martha Palmer\textsuperscript{1} and Nianwen Xue\textsuperscript{2}

\textsuperscript{1} Department of Linguistics,  
University of Colorado,  
Boulder, CO 80302  
martha.palmer@colorado.edu

\textsuperscript{2} Department of Computer Science,  
Brandeis University,  
Waltham, MA 02452  
xuen@brandeis.edu
1 Introduction

The recent availability of linguistically annotated electronic text has revolutionized the fields of Natural Language Processing and Machine Translation. The creation of the Penn Treebank (Marcus et al., 1993) and the word sense-annotated SEMCOR (Miller, 1995; Fellbaum et al., 1998) showed how even limited amounts of annotated data can result in major improvements in complex natural language understanding systems. These and other annotated corpora have led to the training of stochastic natural language processing components which resulted in high-level improvements for parsing and word sense disambiguation (WSD), similarly to the improvements for Part of Speech tagging attributed to the annotation of the Brown corpus and, more recently, the British National Corpus (BNC) (Burnard, 2000). These successes have encouraged the development of an increasingly wide variety of corpora with richer and more diverse annotation. These include the Automatic Content Extraction (ACE) annotations (Named Entity tags, nominal entity tags, coreference, semantic relations and events); semantic annotations, such as more coarse-grained sense tags (Palmer et al., 2007); semantic role labels as in PropBank (Palmer et al., 2005), NomBank (Meyers et al., 2004), and FrameNet (Baker et al., 1998); and pragmatic annotations, such as coreference (Peosio & Vieira, 1998; Poesio, 2004), temporal relations as in TimeBank (Pustejovsky et al., 2003, 2005), the Opinion corpus (Weibe et al., 2005), and the Penn Discourse Treebank (Milstakaki et al., 2004b), to name just a few.

The depth of representation that NLP systems currently aspire to is in fact defined by the availability of corresponding linguistic annotations. For machine learning systems to be trained to produce transformations that add substantially new information to textual input, they have to first be exposed
to similar information in context. In most cases these systems do an admirable job of automatically reproducing the same types of annotation, assuming the annotation has been carried out consistently. The higher the Inter-Annotator agreement, and the greater the consistency and coherence of the original annotation, the higher the probability of acceptable performance of the trained systems. Not surprisingly there is now an insatiable demand for more and more annotated data: the same types of annotations for different genres and different languages; newer, richer annotations for the original languages; parallel annotations of parallel corpora; the merging of annotations that were first done independently; and new formatting for pre-existing annotations that makes them easier to merge. For today’s NLP systems, the annotation defines the task, and increasingly rich annotations are the key to more sophisticated systems. Clearly annotation work needs to become much more widely distributed to cope with this need. The field requires a better understanding of reliable annotation processes for several different types of linguistic annotation that can be readily ported.

It is tempting to assume that recent advances in semi-supervised and unsupervised machine learning (see Chapter 9) may eventually obviate the need for linguistic annotation, but this is not likely. Even unsupervised systems rely on manually annotated data for evaluation purposes. The ready portability of these systems to other genres and languages will simply increase the clamor for additional annotation, albeit in smaller amounts than would be necessary for supervised approaches. In the meanwhile, applications that are aiming at the highest possible accuracy levels continue to rely on supervised machine learning.
In this chapter we first present details of several different specific annotation projects and then review the basic elements that must be considered to achieve consistent annotation, which are generally applicable to different types of annotation. These include:

- Target phenomena definition
- Corpus selection
- Annotation efficiency and consistency
- Annotation infrastructure
- Annotation evaluation
- The use of machine learning for preprocessing and sampling
2 Review of selected annotation schemes

Covering every individual annotation scheme in every language is beyond the scope of this chapter. In this section we review a representative set of widely used resources that range from syntactic annotation to pragmatic annotation, including:

*Syntactic Structure, e.g., Treebanking*

Associating a manual syntactic parse (a complicated structure) with every sentence in a corpus consisting of a set of documents. Whether the target structure is dependency structure or phrase structure, this is an unusually difficult type of annotation that requires in-depth training of annotators. A pre-processing step that involves tokenization, end of sentence detection and part-of-speech tagging is usually involved. Because of the labor intensive nature of treebanking it is usually done with single annotation - one person looks at each sentence.

*Independent Semantic classification, e.g., Sense Tagging*

Based on a pre-existing sense inventory or set of semantic classes, every instance of a specific lemma (a word form corresponding to a unique lexical entry) in a corpus is manually tagged with its relevant sense, or class. This usually involves the same pre-processing as treebanking, including part-of-speech tagging, and is typically done as double blind annotation (two independent taggers) with adjudication of discrepancies. It requires only minimal training.
Semantic Relation Labeling, e.g., Semantic Role Labeling

This is a more complex task, since it involves identifying a target relation and one or more participants in that relation. Semantic role labeling often begins with a corpus of parsed sentences, and the arguments associated with distinct subcategorization frames of verbs are given consistent label names according to a predefined lexical resource of frame descriptions. This is also typically done as double-blind annotation, and can be applied to predicative nouns as well as verbs, or to other types of relations, such as discourse relations and temporal relations. Training must include familiarizing the annotators with the parses, if provided, and the relation descriptions.

Discourse Relations

Since they typically involve relations between sentences or sentence fragments, discourse relations can be viewed as an additional type of semantic relation. For example, the Penn Discourse Treebank (PDTB), funded by NSF, is based on the idea that discourse connectives such as and, but, then, while, ... can be thought of as predicates with associated argument structures (Miltsakaki et al., 2004a).

Temporal Relations

Our final example of semantic relations consists of temporal relations, such as those found in TimeBank. Given a corpus where both nominal and verbal events and their participants have been identified, relations between the events, such as temporal and subordinating relations, are identified and labeled using a predefined set of relationship types.
Coreference Tagging

References to entities in a document are identified as mentions, and mentions of the same entity are linked as being coreferences, or members of a coreference set. These can include pronouns, nominal entities, named entities, elided arguments, and events. Techniques for annotating and evaluating entire sets of coreferences are significantly more complex than techniques for straightforward class-labeling or relation-labeling tasks.

Opinion Tagging

The annotation of opinions, evaluations, emotions, sentiments, and other private states in text is collectively described as Opinion Tagging or Sentiment Tagging. At its simplest this could be seen as a type of semantic classification task, but since the tagging typically includes filling in several different feature values rather than simply assigning class labels, it is discussed separately.

These different types of annotation are all described in more detail below.

2.1 Syntactic Structure, i.e., Treebanking

The dramatic improvement in natural language parsing achieved during the last two decades or so has been generally attributed to the emergence of statistical and machine learning approaches (Collins, 1999; Charniak, 2000). However, statistical and machine learning methods are only possible with the availability of large-scale treebanks, corpora of hand-crafted syntactic trees. The Penn Treebank (PTB) (Marcus et al., 1993) played a special role in providing a shared dataset on which competing parsing approaches are trained and tested. In our view, there are two main factors that contributed to the success of the Penn Treebank. The first one has to do with its size. Although
not the first syntactically annotated corpus, the Penn Treebank is the first one that covers over two million words of text. Statistical approaches to natural language parsing require large quantities of training data to get reliable statistics for the large number of grammatical and lexical phenomena in a language, and this is provided by the Penn Treebank. The one million word Wall Street Journal subcorpus of the PTB is the most frequently used data set even though Wall Street Journal articles are not particularly representative of the English language. The other factor for the PTB’s success is its pragmatic approach. Many key annotation decisions are driven by engineering desiderata rather than purely by linguistic considerations. This often means that theoretically important linguistic distinctions that are hard to make are left unspecified for the sake of annotation consistency. For example, the argument / adjunct distinction has been a key building block for theoretical linguistics, but is avoided in the Penn Treebank. In hindsight, this decision cuts both ways. On the one hand, it simplifies the annotation task and has led to more consistent annotation. On the other hand, it leaves out key information that has to be recovered later in the semantic layer of annotation. For example, the argument/adjunct distinction had to be made at least superficially in the development of the PropBank (Palmer et al., 2005), which adds a layer of predicate-argument annotation to the Penn Treebank.

*Phrase structure treebanks*

The success of the Penn Treebank inspired the development of treebanks in other languages. At Penn, the Chinese (Xue et al., 2005), Korean (Han et al., 2002) and Arabic (Maamouri & Bies, to appear) Treebanks have all been developed using a similar annotation scheme. This annotation scheme is characterized by labeled phrase structures, supplemented by functional tags that
represent grammatical relations such as subject (-SBJ), temporal (-TMP) and locational modifiers (-LOC), as well as empty categories and co-indices that link empty categories to their explicit co-referents, a hallmark of generative grammar. Empty categories and co-indices are used to represent left-displacement, but they are by no means the only way to represent movement phenomena. The different aspects of the representation scheme are illustrated in Figure 1: *The other half* is an NP with the functional tag -TPC, indicating it is an NP playing the role of a topic, a grammatical position in English syntax. This topic originates in the object position of the verb *have* and then moves to the sentence-initial position. This information is represented by the empty category *T* in the object position co-indexed with the topic in the sentence-initial position. The empty category and co-indexation mechanism localize the arguments for a predicate and thus make it easier to extract the predicate-argument structure. *The other half* is made adjacent to the verb *have* by positing an empty category *T* that is co-indexed with it. Although *head* is a prominent notion in generative grammar, it is not explicitly represented in the Penn Treebank. However, the notion of head is implicitly built-in in the structural configuration of a phrase and in principle can be identified via a finite set of rules defined for each syntactic category. For example, the verb *have* is assumed to be the head of the VP *have *T*-1 before long* by virtue of being the first verb in this VP. This set of rules are generally referred to as a head table and are widely referenced in statistical parsing literature. In practice, due to annotation errors and underspecified annotation, the head cannot always be reliably identified. Therefore, in some phrase structure annotation schemes, the head is explicitly marked to avoid such pitfalls.
when extracting the head. For example, the Tiger corpus for German (Brants et al., 2002) explicitly marks the head of a phrase.

![PTB Tree](image)

**Figure 1.** An example PTB Tree

*Dependency treebanks*

The Prague Dependency Treebank (Hajič, 1998) represents a radically different annotation scheme in the Functional Generative Description framework, following the Prague dependency tradition. At the core of this annotation scheme is the dependency relation between a head and its dependent. While in a phrase structure representation the explicit markup of the head is optional, identifying the head is essential in a dependency structure. The dependency relation between a head and its dependent is the building block of the dependency structure of a sentence. A dependency structure representation of the same sentence as in Figure 1 is provided in Figure 2. While in a PTB-style phrase structure the dependency between the verb *have* and the topic *the other half* is mediated via a co-indexation mechanism, in the dependency structure this relation is represented directly. Another key difference is that while the syntactic categories of constituents in a phrase structure tree represent the dis-
tributional properties of the constituents, e.g., noun phrases generally occur in subject and object positions, etc., the focus of a dependency representation is on the relation between a head and its dependent. Therefore, while the nodes in a dependency tree are labeled by the head, which is not particularly informative other than saying the parent is the head and the child is the dependent, the edges are often labeled with dependency relations such as subject and object. For example, there is a SBJ relation between have and we, and a TPC relation between have and the other half. As Xia & Palmer (2001) showed, since there are no phrasal labels in a dependency representation, and more importantly, the subject and complement are all attached to the head at the same level, it is generally not possible to automatically convert a dependency tree such as the one in Figure 2 to the PTB-style phrase structure representation. On the other hand, assuming that the head can be reliably identified, it is possible to automatically derive this dependency structure representation from a phrase structure representation. However, since there is no limit to the possible labels for dependency relations, it might be possible to label dependency relations in such a way that a phrase structure representation can be reconstructed.

Figure 2. A labeled dependency structure
There is a growing realization that both dependency and phrase structure treebanks are needed. In fact, both Tree-Adjoining Grammar (TAG) and Lexical Functional Grammar (LFG) provide in a sense a combination of phrase structure and dependency structure representations, with the LFG c-structure (constituent structure) corresponding to the phrase structure layer and the LFG f-structure (functional structure) corresponding to the dependency structure layer. With TAG the derivation tree is the constituent structure and the derived tree is closer to the dependency structure. Although there are no manually annotated large-scale LFG style or TAG style treebanks that we are aware of, there have been efforts to convert the phrase structure annotation of the Penn Treebank into both TAG structures (Fei Xia & Joshi, 2001) and LFG f-structures (Cahill et al., 2002), and this has provided training data for successful statistical TAG and LFG parsers. There is a recent initiative funded by NSF to build a Hindi/Urdu dependency treebank that has rich enough annotation that it can be readily converted to a phrase structure treebank Bhatt et al. (2009).

### 2.2 Semantic Classification, i.e., Sense Tagging

Sense tagging is essentially a semantic classification task. Given a set of pre-defined semantic labels for a distinct lexical item, one or more labels is associated with each occurrence of that item in a sentential context in a corpus. For instance, the *call* lemma in (1) is tagged with the OntoNotes (See Section 2.8) Sense 1 for *call*, corresponding to *communicate orally, usually in a loud, distinct tone*.

(1) "You people here think this is Russian music, "she said with disdain, and called over to the waitress: “Could you turn it off?"
In contrast, (2) is tagged with Sense 5, to label with a name or quality.

(2) “A spokesman for the state, however, calls the idea “not effective or cost efficient.””

The traditional assumption is that the labels correspond to sense entries from a pre-existing sense inventory, such as a dictionary, and annotators apply these labels after reading the sentence containing the lemma.

There are other closely related tasks such as nominal entity tagging which are also basically semantic classification tasks but with different notions of where the class labels come from. Nominal entity tagging, as defined by the Automatic Content Extraction (ACE) project (Strassel et al., 2008), is focused primarily on nouns and consists of choosing a semantic category from a predefined category list (PERson, ORGanization, GeoPoliticalEntity, Location, FACility, SUBstance, VEHicle, WEApion) for each occurrence of the noun in context in a corpus. Several nouns, especially proper nouns such as the White House, can have multiple tags, such as PER, GPE, or LOC. In these cases, determining which tag is appropriate given a specific sentence as the context amounts to the equivalent of a sense-tagging task. An important difference is that for nominal entity tagging there is one set of sense tags for all nouns, rather than a unique set of sense tags for each lexical item. However, the entity type tags can easily be mapped to standard dictionary sense entries, which for nouns in particular are often separated according to semantic category. For instance, in the following example the definition of regulator has two senses, one of which can be mapped to the SUBstance category and the other of which can be mapped to the ORGanization or PERson category. If a corpus containing this word had already been annotated with nominal entity

---

\footnote{The ACE guidelines include complex subtypes for each of these categories and multitudinous examples.}
categories, then those category labels for regulator could be deterministically mapped to either Sense 1 or Sense 2, providing a sense-tagged corpus.

regulator

1: a device to control the rate of some activity, e.g., chemical or mechanical - SUBSTANCE

2: an official with responsibility to supervise some domain of social activity - ORG, PER

Choosing the sense inventory

Clearly the most important decision in creating sense-tagged corpora (or nominal entity tagged corpora) is the choice of the sense inventory (or label set) that will be used. For classic sense tagging tasks this is a computational lexicon or a machine readable dictionary that partitions the meaning of each word into numbered senses, allowing the word plus a number to uniquely refer to an entry. An ideal sense inventory should make clear and consistent sense distinctions for each word. Unfortunately, sense inventories for a language can be discouragingly diverse, with significant differences with respect to entries for polysemous words, and different levels of granularity of the sense distinctions. Corpora tagged with two different English sense inventories will not provide coherent training data unless a comprehensive mapping can be provided between every entry, and the mappings are often not one-to-one (Senseval2).

The sense-tagged data itself can be framed as either lexical sample data or as an all-words corpus. In the all-words corpus, all words (or all content words) in a running text or discourse are tagged. While superficially similar to
part-of-speech tagging,\textsuperscript{2} all-words tagging via a sense inventory is significantly different in that a different set of sense tags is required for each lemma. If public distribution is desired, this severely limits the choice of possible sense inventories, because it requires access to a publicly available, wide-coverage dictionary that is preferably also free or at least low-cost. For a lexical sample corpus, a sample of words is carefully selected from the lexicon, along with a number of corpus instances of each word to be tagged. Unlike all-words tagging, dictionary entries are required only for these select words, so given a small enough sample of words, there could be more flexibility in dictionary choice.

The methodology for manual annotation depends on the type of tagging. Words can be annotated more quickly and consistently if all instances of a word (type) are tagged at once (targeted tagging), instead of tagging all words sequentially as they appear in the text. The advantages of targeted tagging make lexical sample tagging easier to implement than all-words tagging. The largest all-words corpus, SemCor, based on the Brown corpus (Francis & Kucera, 1982), is tagged with WordNet senses (Miller, 1995). Created by George Miller and his team at Princeton University, WordNet (Miller \textit{et al.}, 1990; Miller & Fellbaum, 1991; Fellbaum \textit{et al.}, 1998) is a large electronic database organized as a semantic network built on paradigmatic relations including synonymy, hyponymy, antonymy, and entailment. WordNet has become the most widely used lexical database today for NLP research, and its approach has now been ported to several other languages, such as several European languages in EuroWordNet (Vossen, 1998) and in BalkaNet (Stamou \textit{et al.}, 2002), as well a Japanese (Bond \textit{et al.}, 2008) and a Chinese WordNet (Xu \textit{et al.}, 2008).}

\textsuperscript{2} Named entity (or nominal entity) tagging is very similar to part-of-speech tagging, in that one set of tags is used for all entities.
WordNet lists the different senses for each English open-class word (nouns, verbs, adjectives, and adverbs). Sense tagging is typically done as double-blind annotation by two linguistically or lexicographically trained annotators, with a third tagger adjudicating between inter-annotator differences to create a Gold Standard. Because of issues that have been raised about low ITA rates due to fine-grained sense distinctions in English WordNet, and corresponding unacceptably low system performance, manual groupings of WordNet senses are now being used for tagging in a large DARPA-funded project (see Section 2.8.)

2.3 Semantic Relation Labeling, i.e., Semantic Role Labeling

A closely related but distinct semantic annotation task involves identifying within a single sentence a relation and its arguments, and then labeling the arguments. The classic example is a verb where the labels of the verb arguments, once they have been identified, correspond to semantic role labels. The semantic role labels are intended to indicate a specific semantic relation between a verb and its argument that holds consistently even when the argument is in different syntactic positions. This description covers several current semantic role labeling tasks, in which the semantic roles can come variously from PropBank, (Palmer et al., 2005), FrameNet (Baker et al., 1998), VerbNet (Kipper et al., 2006), or the Prague Tecto-Grammatical formalism (Hajič et al., 2000). It can also be extended to similar semantic roles that are introduced by other parts of speech, such as nominal or adjectival elements.

A major difference between sense tagging and semantic role labeling is the interdependence between the semantic role labels. If the Agent, or Arg0, of a verb has already been labeled, that changes the available choices for the remaining arguments. This interdependence of labels is also the case for dis-
course relations and temporal relations; however, given the distinctive nature of these annotation tasks, they will be dealt with in separate sections. ACE relations, which are also similar, are discussed at the end of this section.

*The Proposition Bank*

The Proposition Bank, originally funded by ACE (DOD), focuses on the argument structure of verbs, and provides a corpus annotated with semantic roles, including participants traditionally viewed as arguments and adjuncts. Correctly identifying the semantic roles of the sentence constituents, or *Who did what to whom, and when, where and how?* is a crucial part of interpreting text, and in addition to forming a component of the information extraction problem, can serve as an intermediate step in machine translation, automatic summarization or question answering.

At the beginning of the PropBank project, the decision was made to associate the semantic role labels directly with nodes in the Penn Treebank phrase structure parses. The boundaries of the constituents corresponding to the nodes were already defined, so the annotators did not have to add that task to their duties, simplifying their cognitive load. The PTB also had indicated empty arguments, so these could easily be given semantic role labels as well, making the annotation more complete. Finally, the assumption was that the syntax and semantics would be highly correlated, with the semantic roles occurring within the domain of locality of the predicking element. Therefore having access to the syntactic structure should help in simplifying and focusing the task for the annotators. In contrast, FrameNet (see below) annotation did not initially begin with pre-parsed sentences, and this was found to lower agreement among the annotators, primarily because of different constituent boundary decisions. Another feasible method is to annotate depend-
ency structure parses directly with semantic role labels, and the Hindi/Urdu Treebank project will explore this approach in depth. One of the questions to be addressed is the issue of empty arguments; if empty arguments have not been inserted by the dependency annotation, should they be added at the PropBank level? Since the goal of this project is to eventually convert the dependency structure + PropBank annotation automatically into a phrase structure treebank, having the empty arguments in place could simplify the conversion process (Bhatt et al., 2009).

The 1M word Penn Treebank II Wall Street Journal corpus has been successfully annotated with semantic argument structures for verbs and is now available via the Penn Linguistic Data Consortium as PropBank I (Palmer et al., 2005). More specifically, PropBank annotation involves three tasks: argument labeling, annotation of modifiers, and creating co-reference chains for empty arguments. The first goal is to provide consistent argument labels across different syntactic realizations of the same verb, as in

(3) a) “[ARG0 John] [REL broke] [ARG1 the window]”
   b) “[ARG1 The window] [REL broke].”

The Arg1 or PATIENT in (3aa) is the same window that is annotated as the Arg1 in (3ab), even though it is the syntactic subject in one sentence and the syntactic object in the other. As this example shows, semantic arguments are tagged with numbered argument labels, such as Arg0, Arg1, Arg2, where these labels are defined on a verb-by-verb basis. The second task of the PropBank annotation involves assigning functional tags to all modifiers of the verb, such as MNR (manner), LOC (locative), TMP (temporal), DIS (discourse connectives), PRP (purpose) or DIR (direction) and others, as in (4).
Finally, PropBank annotation involves finding antecedents for empty arguments of the verbs, as in (5).

(5) “You people here think this is Russian music, “she said [*T*-1] with disdain, and called over to the waitress: “Could you turn it off?””

The object of the verb *say* in this example, *You people here think this is Russian music* is represented as an empty category [*T*-1] in Treebank. In PropBank, all empty categories that could be co-referred with an NP within the same sentence are linked in co-reference chains. So the [*T*-1] is linked to *You people here think this is Russian music*. The primary goal of PropBank is to supply consistent, simple, general purpose labeling of semantic roles for a large quantity of coherent text that can provide training data for supervised machine learning algorithms, in the same way the Penn Treebank has supported the training of statistical syntactic parsers. PropBank also provides a lexicon which lists, for each broad meaning of each annotated verb, its Frameset, i.e., the possible arguments in the predicate and their labels and all possible syntactic realizations. PropBank’s focus is verbs, so NomBank, annotation of nominalizations and other noun predicates using PropBank style Framesets, was done at NYU, also funded by ACE Meyers et al. (2004).

FrameNet

FrameNet consists of collections of semantic frames, lexical units that evoke these frames, and annotation reports that demonstrate uses of lexical units. Each semantic frame specifies a set of frame elements, or arguments. Semantic frames are related to one another via a set of possible relations such as is-a and
uses. Frame elements are classified in terms of how central they are to a particular frame, distinguishing three levels: core, peripheral, and extra-thematic. FrameNet is designed to group lexical items based on frame semantics, and sets of verbs with similar syntactic behavior may appear in multiple frames, while a single FrameNet frame may contain sets of verbs with related senses but different subcategorization properties. FrameNet places a primary emphasis on providing rich, idiosyncratic descriptions of semantic properties of lexical units in context, and making explicit subtle differences in meaning. As such it could provide an important foundation for reasoning about context-dependent semantic representations. However, the large number of frame elements and the current sparseness of available annotations for each one has been an impediment to machine learning.

VerbNet

VerbNet is midway between PropBank and FrameNet in terms of lexical specificity, and is closer to PropBank in its close ties to syntactic structure. It consists of hierarchically arranged verb classes, inspired by and extended from classes of Levin 1993. Each class and subclass is characterized by its set of verbs, by a list of the arguments of those verbs, and by syntactic and semantic information about the verbs. The argument list consists of thematic roles (23 in total) and possible selectional restrictions on the arguments expressed using binary predicates. Additional semantic predicates describe the participants during various stages of the event described by the syntactic frame, and provide class-specific interpretations of the thematic roles. VerbNet now covers over 6000 senses for 5319 lexemes. A primary emphasis for VerbNet is the coherent syntactic and semantic characterization of the classes,
which will facilitate the acquisition of new class members based on observable syntactic and semantic behavior.

SemLink

Although PropBank, FrameNet and VerbNet have all been created independently, with differing goals, they are surprisingly compatible in their shared focus on labeling verb arguments. PropBank uses very generic labels such as Arg0, as in (6).

(6) “[ARG0 President Bush] has [REL approved] [ARG1 duty-free treatment for imports of certain types of watches].”

In addition to providing several alternative syntactic frames and a set of semantic predicates, VerbNet marks the PropBank Arg0 in this sentence as an Agent, and the Arg1 as a Theme. FrameNet labels them as Grantor and Action respectively, and puts them in the Grant.Permission class. The additional semantic richness provided by VerbNet and FrameNet does not contradict PropBank, but can be seen as complementary. These resources can also be seen as complementary with WordNet, in that they provide explicit descriptions of participants and ties to syntactic structure that WordNet does not provide. The PropBank labels, being the most generic, will cover the widest number of WordNet senses for a particular word. A verb in a VerbNet class will also usually correspond to several WordNet senses, which are explicitly marked. FrameNet provides the finest sense granularity of these resources, and specific FrameNet frames are more likely to map onto individual WordNet senses. There are significant differences in the coverage of lexemes and the structuring of data in each of these resources, which could be used to bootstrap coverage extensions for each one. The simple labels provided by PropBank are more
amenable to machine learning, and have resulted in the training of successful automatic semantic role labeling systems. A semi-automatic mapping from PropBank to VerbNet has been produced (and hand-corrected) which has been used to successfully train systems that can produce either PropBank or VerbNet semantic role labels (Yi et al., 2007). 3465 types have been mapped, comprising over 80% of the tokens in the PropBank. In parallel a type-to-type mapping table from VerbNet class(es) to FrameNet frame(s) has been created, as well as a mapping from role label to frame element. This will facilitate the generation of FrameNet representations for every VerbNet version of a PropBank instance that has an entry in the table.

ACE Relations

This style of annotation also bears a close resemblance to the ACE Relation task, which is aimed at detecting within a sentence a particular type of relation and its arguments (LDC, 2008). There is a shift in focus with ACE, however, from a lexically oriented, linguistically motivated task, such as semantic role labeling, to a more pragmatic, relation-type task. The relation types include: PHYSICALLY located, as in LOCATED or NEAR (see (7); PART-WHOLE, which could be a GEOGRAPHICAL PART-WHOLE relation, such as Colorado being PART of the United States, or SUBSIDIARY, as in Umbria being a SUBSIDIARY or PART of JD Powers; PERSONAL-SOCIAL, as in BUSINESS (co-workers), FAMILY (siblings), or LASTING-PERSONAL (life-long neighbors); ORG-AFFILIATION, which includes EMPLOYMENT, OWNERSHIP, FOUNDER, etc.; AGENT-ARTIFACT, and others. For example, there is a PHYSICALLY Located relation between ”Barack Obama” and ”Germany” in (7). An important distinction is that the same ACE relation type could be
introduced by a verb, a noun or a preposition, so the annotators need to focus more on semantic content and less on syntactic structure.

(7) “Barack Obama traveled to Germany to give a speech at Buchenwald.”

2.4 TimeBank

TimeBank (Pustejovsky et al., 2003) is a corpus annotated with temporal information based on TimeML (Pustejovsky et al., 2005), a general purpose temporal markup language that has been adopted as an ISO semantic annotation standard (ISO/TC 37/SC 4/WG 2, 2007). The basic elements of the TimeML are events, time expressions, and signals, as well as temporal relations between these temporal entities. For example, in (8), glossed, warnings, strikes, do and harm would all be identified as anchors of events in the TimeBank annotation; Thursday would be marked up as a temporal expression; and on would be marked as a signal for the temporal relation between the glossing-over event and the temporal expression Thursday. Temporal relations also hold between events. For example, the strike event would precede the harm event, which would in turn be annotated as identical to the do event.

(8) “President Clinton, meantime, glossed over stern warnings from Moscow on Thursday that US air strikes against Iraq could do serious harm to relations with the Kremlin.”

TimeML adopts a broad definition of event. Event for TimeBank is a cover term for situations that happen or occur. Events can be punctual or last for a period of time. The TimeBank events also include states or circumstances in which something obtains or holds true. However, TimeBank does not mark up all states in a document. It only annotates states that are
relevant to temporal interpretation, for example, states that are identifiably changed during the document time (9a), or states (9b) that are directly related to a temporal expression. States that persistently hold true are excluded from annotation. Syntactically, events can be realized as verbs, nominalizations, adjectives, predicative clauses, or prepositional phrases.

(9) a) “All 75 on board the Aeroflot Airbus died.”
   b) “They lived in U.N.-run refugee camps for 2 2/1 years.”

A time expression belongs to one of the four types: Date, Time, Duration or Set. A Date describes a calendar time, and examples are Friday, October 1, 1999, yesterday, last week. Time refers to a time of the day, even if it is indefinite, e.g., ten minutes from three, five to eight, late last night. Durations are assigned to explicit durations such as 2 months, and 48 hours. Finally, a Set describes a set of times, e.g., twice a week or every two days. Time expressions are also annotated with a normalized value. For example, twelve o’clock midnight would be normalized to T24:00.

A signal is a textual element that makes explicit the temporal relation between a temporal expression and an event, or between a temporal expression and a temporal expression, or between an event and an event. Signals are generally temporal prepositions such as on, in, at, from, to, or temporal conjunctions such as before, after, while, when, as well as special characters like “-” and “/” that indicate ranges of time.

Events, time expressions and signals are temporal entities that are linked by temporal relations to form an overall temporal interpretation of a text. The main temporal relations are represented by Temporal Links (TLINKs), which represent the temporal relation between events, between times, or between an event and a time. The TLINK annotation is illustrated in (10), where there is
a BEFORE relation between the events anchored by *invited* and *come*, which means the inviting event happens before the coming event.

(10) “Fidel Castro invited John Paul to *come* for a reason.”

Subordination Links (SLINKs) are another type of temporal link, and they are used to represent modal, factive, counter-factive, evidential, and negative evidential relations, as well as conditional relations that usually hold between a main event and a subordinate event. For example, in (11), an SLINK can be established between the events anchored by *adopt* and *ensure*.

(11) “The Environmental commission must *adopt* regulations to *ensure* people are not exposed to radioactive waste.”

A third and final type of link is Aspectual Link ALINK, which represents the relationship between an aspectual event and its argument event. The relation that an ALINK represents can be one of five types: Initiates, Culminates, Terminates, Continues, or Reinitiates. The example in (12) represents an Initiates relation between the events anchored by *began* and *trading*.

(12) “The stock *began* trading this summer at $14 apiece.”

Achieving consistency in the TimeBank annotation has proven to be very difficult, with temporal ordering of events being the most challenging part of the annotation. It is neither feasible nor necessary to temporally order each pair of events in a document, but without some clear guidance, different annotators tend to choose different pairs of events to annotate, leading to poor inter-annotator agreement. In practical temporal annotation, some form of temporal inference mechanism has to be implemented (Verhagen, 2005) so that the temporal ordering of some pairs of events can be automatically
inferred. The fine-grained nature of some temporal relations also makes it difficult to separate one relation from another. simultaneous or

2.5 Discourse relation annotation

Treebank and PropBank annotations are all focused on getting linguistic information from within the sentence. More recently, there have been efforts to annotate linguistic structures beyond the sentence level. These new efforts make the structure of a whole text their target of annotation. In this subsection we discuss two such projects, the RST Corpus and the Penn Discourse Treebank Project, which have taken very different approaches to discourse structure annotation.

RST Corpus

The RST Corpus (Carlson et al., 2003) consists of 385 articles from the Penn Treebank, representing over 176K words of text. The RST Corpus is hierarchically annotated in the framework of Rhetorical Structure Theory (Mann and Thompson, 1988). In Rhetorical Structure Theory, the discourse structure of a text is represented as a tree, and the leaves of the tree are text fragments that represent the minimal units of discourse, called elementary discourse units or EDUs. Each node in the discourse tree is characterized by a rhetorical relation that holds between two or more adjacent nodes, and corresponds to contiguous spans of text. The rhetorical relation between the children of a node is characterized by nearcality, with the nucleus being the essential unit of information, while a satellite indicates a supporting or background unit of information.

The annotation of the RST Corpus starts off by identifying the elementary discourse units, or EDUs, which are building blocks of a discourse tree. The
EDUs roughly correspond to clauses, although not all clauses are EDUs. For example, a subordinate clause that is an adjunct to the main clause is usually an EDU, but clauses that are subjects, objects, or complements of a main clause are not usually EDUs.

The discourse relations between child discourse units of a node in the discourse tree can be either mononuclear or multinuclear, based on the relative salience of the discourse units. A mononuclear relation is between two discourse units where one is the nucleus and another is the satellite. The nucleus represents the more salient or essential information while the satellite indicates supporting and background information. A multinuclear relation is between two or more discourse units that are of equal importance and thus are all nuclei. This in a way parallels the endocentric and exocentric structures at the sentence level. A total of 53 mononuclear and 25 multinuclear relations are used to annotate the RST corpus; these 78 relations fall into 16 broad classes. These discourse relations are identified empirically, based on evidence from the corpus. “Elaboration” is an example of a mononuclear discourse relation, while “list” is a multinuclear discourse relation. For the complete set of discourse relations tagged in the RST, the reader is referred to the discourse tagging manual of the RST corpus.

The Penn Discourse Treebank

While the RST annotation of discourse relations is organized around EDUs, the building blocks of the Penn Discourse Treebank (Miltsakaki et al., 2004a) are discourse connectives and their arguments. The annotation framework of the Penn Discourse Treebank is based on a theoretical framework developed in (Webber and Joshi, 1998), where discourse connectives are considered to be predicates that take abstract objects such as events, states and propositions
as their arguments. The Penn Discourse Treebank annotates both *explicit* and *implicit* discourse connectives and their arguments. Explicit discourse connectives include subordinating conjunctions and coordinating conjunctions, as well as discourse adverbials. While in most cases the arguments of discourse connectives are local and adjacent to the discourse connective, they do not have to be. Webber et al. (Webber & Joshi, 1998) considers subordinating and coordinating conjunctions to be *structural* in the sense that their arguments are local to the discourse connective, while discourse adverbials are considered to be *anaphorical*, because their first arguments can be long-distance.

Where explicit discourse connectives are absent, implicit discourse connectives are inserted between paragraph-internal sentence pairs, as illustrated in (13). In some cases, it may not be possible to insert a discourse connective because the discourse relation is expressed with a non-discourse connective element, or because discourse coherence is achieved by an entity chain, or simply because there is no relation of any kind.

(13) “Motorola is fighting back against junk mail. [ARG1 So much of the stuff poured into its Austin, Texas, offices that its mail rooms there simply stopped delivering it]. Implicit=so [ARG2 Now, thousands of mailers, catalogs and sales pitches go straight into the trash].”

The lexically grounded approach of the Penn Discourse Treebank opens the door for the possibility that one discourse connective might be a lexical realization of multiple discourse relations. The Penn Discourse Treebank addresses this by specifying an inventory of discourse relations that serve as senses of the discourse connectives. In a way, this inventory is similar to the set of discourse relations adopted by the RST, while the actual discourse relations posited might be different. Like the RST Corpus, the discourse re-
Linguistic Annotation 29

relations are hierarchically organized. The top level has four major semantic classes: TEMPORAL, CONTINGENCY, COMPARISON and EXPANSION. For each class, a second level of type is defined, and then for each type, there may be a third level of subtype defined. There are 16 types and 23 subtypes. Some types do not have subtypes. The reader is referred to the PDTB annotation manual for details.

A comparison of the two approaches

The RST Corpus and the Penn Discourse Treebank represent very different approaches to the annotation of discourse relations. The most fundamental difference is that the RST Corpus is committed to building a discourse tree representation for the entire text. The leaves of the tree are elementary discourse units or EDUs, which are non-overlapping spans of text. Discourse units built on the EDUs are also non-overlapping spans of text, and discourse relations are always local in the sense that they only hold between adjacent discourse units. The Penn Discourse Treebank, on the other hand, is not committed to a tree representation of the entire text. The building blocks are discourse connectives, which are treated as predicates, the arguments of which are identified for each discourse connective instance in a text, whether they are explicit or implicit. Although the text spans that are identified as arguments of a discourse connective never overlap, there is no guarantee that arguments of different connectives do not overlap. In fact, (Lee et al., 2006) show that there is a variety of possible patterns of dependencies between pairs of discourse relations, including nested, crossed and other non-tree-like configurations. Part of the reason for this complex structure is due to the anaphoric discourse relations for discourse adverbials, whose arguments are not necessarily local, as discussed above.
Another reason why there exist complex dependencies in the Penn Discourse Treebank is the way attribution is annotated. PDTB adopts the strict view that discourse relations are between abstract objects such as events, states, and propositions. Since attributions are relations between an agent and a proposition, the attribution annotation is treated as a separate layer of annotation. Depending on the context, the attribution may be included as part of an argument in some cases but excluded as others. (14) is an example. The higher verb “He said” is included as an argument of ALTHOUGH, but excluded as an argument of ALSO. There is only a partial overlap between the arguments for these two discourse connectives. If the attribution is excluded from the discourse processing, then the discourse relation for ALTHOUGH would be properly contained as an argument for ALSO. RST, on the other hand, includes attribution as one type of discourse relation, not distinguished from other discourse relations. The RST Corpus also does not consider ALSO as a trigger for a discourse relation. Discourse connectives are used as cues to identify EDUs and determine their discourse relations, but they have no formal role in the RST annotation scheme.

(14) “He (Mr. Meek) said he evidence pointed to wrongdoing by Mr. Keating “and others,” ALTHOUGH he didn’t allege any specific violation. Richard Newsom, a California state official who last year examined Lincoln’s parent, American Continental Corp., said he ALSO saw evidence that crimes had been committed.”

A final difference between the RST annotation and the PDTB annotation is that PDTB only considers discourse relations between clauses, while RST also considers discourse relations between subclause relations. For example,
EDUs in the RST corpus can be phrases. A discourse relation can be between the head NP and its postmodifiers.

Measuring annotation consistency for discourse annotation can be very complicated. Carlson et al. (2003) reports inter-annotator agreement on four levels of the RST Corpus: elementary discourse units, hierarchical spans, hierarchical nuclearity and hierarchical relation assignments. The agreement is the highest for the identification of EDUs (0.97 Kappa) and the lowest in discourse relation assignment (0.75 Kappa), which is not unexpected. Miltsakaki et al. (2004a) reported inter-annotator agreement on the Penn Discourse Treebank using two different measures. The first measure is more lenient and is calculated on a per-argument basis. That is, if two annotators assign the same argument label to the same spans of text, it is counted as a match, regardless of whether the other argument of the same discourse connective is a match or not. By this measure, the average agreement score is 90.2%. The second measure is more stringent and is calculated on a per-discourse relation basis. That is, two annotators have to agree on both arguments of a discourse connective in order for that to be counted as a match. By this measure the agreement is 82.8%. Although the consistency measures are not comparable between the two discourse annotation projects, it appears that both projects have achieved reasonable consistent scores, indicating the viability of these annotation schemes (see Section 3.5).

2.6 Coreference

References to entities in a document are identified as mentions, and mentions of the same entity are linked as being coreferences, or members of a coreference set. These can include pronouns, nominal entities, named entities, elided arguments, and events. For example, Barack Hussen Obama II and he in
Researchers at Essex (UK) were responsible for the coreference markup scheme developed in MATE (Poesio et al., 1999; Poesio, 2004), partially implemented in the annotation tool MMAX and now proposed as an ISO standard; and have been responsible for the creation of two small, but commonly used anaphorically annotated corpora: the Vieira / Poesio subset of the Penn Treebank (Peosio & Vieira, 1998), and the GNOME corpus (Poesio, 2004). Their work also includes extended guidelines (Mengel et al., 2000), and annotation of Italian. Parallel coreference annotation efforts funded first by ACE and more recently by DARPA GALE (see Section 2.8) have resulted in similar guidelines, best exemplified by BBN’s recent efforts to annotate Named Entities, common nouns and pronouns consistently (Pradhan et al., 2007c). These two approaches provide a suitable springboard for an attempt at achieving a community consensus on coreference.

“Barack Hussein Obama II is the 44th and current President of the United States. He is the first African American to hold the office.”

Techniques for annotating and evaluating entire sets of coreferences are significantly more complex than techniques for straightforward class labeling tasks.

2.7 Opinion annotation

The Pittsburgh Opinion annotation project (Weibe et al., 2005) funded by IARPA, focuses on the annotation of opinions, evaluations, emotions, sentiments, and other private states in text. A fine-grained annotation scheme has been developed for annotating text at the word and phrase levels. For every expression of a private state, a private state frame is defined that identifies
whose private state it is, what the private state is about, and various properties involving intensity, significance, and type of attitude. For example, in (16) a private state frame is anchored by *fear*. The source of the private state is attributed to *the US*, and the attitude type of the private state is *negative*. The intensity of the private state is *medium*. A corpus of over 15,000 sentences has been annotated according to the scheme. The corpus is freely available at: nrrc.mitre.org/NRRC/publications.htm.

(16) “The US fears a spill-over.”

There are several applications for corpora annotated with rich information about opinions. Government, commercial and political information analysts are all interested in developing tools that can automatically track attitudes and feelings in the news and in on-line forums. They would also be interested in tools that would support information extraction systems trying to distinguish between factual and non-factual information as well as question answering systems that could present multiple answers to non-factual questions based on opinions derived from different sources. In addition there is an interest in multi-document summarization systems, which would summarize differing opinions and perspectives.

2.8 Multi-layered annotation projects

The annotation resources we have described so far in this section are mostly one-dimensional tasks that focus on a single language processing goal. Treebanks are used to develop syntactic parsers and propbanks are used to train semantic role labelers. A recent trend in linguistic annotation is aimed at building multi-layered linguistic resources, fueled by the realization in the
natural language processing community that there is great value in annotating the same linguistic source with multiple levels of linguistic information. A major advantage of having a multi-layered linguistic resource is that information encoded in one layer of representation can be used to infer that of another. For example, the role of syntactic parsing in semantic role labeling, a form of semantic parsing, is well-documented (Gildea & Palmer, 2002; Punyakanok et al., 2005). It is perhaps not a coincidence that many semantic annotation projects are built on top of syntactic annotation projects, as discussed in Section 2.3. For example, PropBank (Palmer et al., 2005) is built on top of the Penn Treebank (Marcus et al., 1993). The Salsa Project (Burchardt et al., 2006), a semantic annotation project for German, is built on top of the Tiger Treebank (Brants et al., 2002), a syntactically annotated corpus. The Prague Dependency Treebank has a syntactic (the analytical layer) and semantic annotation layer (the tectogrammatical layer). Perhaps the most ambitious multi-layered annotation project is OntoNotes (Pradhan et al., 2007a), funded through GALE, a large-scale DARPA program focused on automatic machine translation and summarization of Arabic and Chinese speech and text. OntoNotes is a five-year, multi-site collaboration between BBN Technologies, the Information Sciences Institute of the University of Southern California, the University of Colorado, the University of Pennsylvania and Brandeis University. The goal of the OntoNotes project is to provide linguistic data annotated with a skeletal representation of the literal meaning of sentences including syntactic parse, predicate-argument structure, coreference, and word senses linked to an ontology, allowing a new generation of language understanding technologies to be developed with new functional capabilities. The OntoNotes annotation covers multiple genres (newswire, broadcast news, broadcast
conversation and weblogs) in multiple languages (English, Chinese and Arabic). The guiding principle has been to find a “sweet spot” in the space of inter-tagger agreement, productivity, and depth of representation. Figure 3 illustrates the inter-connection between the different layers of linguistic annotation in OntoNotes.

Many new challenges come with multi-layered annotation, particularly for annotation models like OntoNotes. Since the different layers of annotation are performed in different sites following different guidelines, incompatibilities can arise. (17) is an example with the same sentence annotated with both syntactic parses and semantic roles. By assigning different semantic roles to a letter (Arg1) and for Mary (Arg2), the PropBank annotator makes the implicit judgment that for Mary is an argument to the verb wrote and should be attached to this verb instead of the noun phrase a letter, a judgment that is different from the treebank annotation where the PP for Mary is treated as a modifier of a letter. In order to achieve coherent annotation, these
incompatibilities need to be reconciled. Babko-Malaya et al. (2006) describe the many inconsistencies between syntactic parses and predicate-argument structure annotation that need to be resolved under the OntoNotes annotation effort.

(17) a) “(S (NP She)(VP wrote (NP (NP a letter)(PP for Mary))))”

b) “[Arg0 She] wrote [Arg1 a letter] [Arg2 for Mary]”

A linguistic source with multiple linguistic annotation also accentuates the data access problem. The most effective use of such a resource requires simultaneous access to multiple layers of annotation. OntoNotes addresses this by storing the corpus as a relational database to accommodate the dense connectedness of the data and ensure consistency across layers. In order to facilitate ease of understanding and manipulability, the database has also been supplemented with an object-oriented Python API (Pradhan et al., 2007a).
3 The Annotation Process

Linguistic annotation is still in its infancy, and only a small portion of possible annotation schemes have been clearly defined and put into practice. The creation of each new level of annotation involves equal amounts of linguistic knowledge, inspiration and experimentation: linguistic knowledge of the phenomena that need to be identified and described as a justifiable layer of annotation; inspiration as to achievable levels of granularity and precision; and experimentation to determine the gaps and weaknesses in the guidelines. For any particular scheme a finite list of allowable annotations must be specified, with careful attention being paid to accounting for all possible contexts and to the cognitive load on the annotators. A clear understanding of linguistic phenomena does not necessarily translate directly into the development of annotated data that is suitable for training machine learning systems. The more precise the description of the linguistic phenomena is, the greater the likelihood of a sparse data problem. Issues with respect to consistency, coherence and clarity of the annotation scheme need to be well understood; a goal that can only be attained through trial implementations. From the perspective of the annotators, the ideal annotation scheme is clear and unambiguous in all circumstances and can be learned quickly by someone without an extensive linguistic background. This absolute goal may be unattainable, but efforts in its direction are rewarded by rapid, consistent annotation.

The development of an annotation scheme requires addressing at a minimum the following issues, each of which will be discussed in turn below:

- Target phenomena definition
- Corpus selection
- Annotation efficiency and consistency
3.1 The phenomena to be annotated

The most important decision has to do with the phenomena to be annotated. Annotation can be an extremely expensive and labor intensive process, so the first question is, is it absolutely essential that this phenomena be manually annotated, or will automatic techniques be almost as good? Having decided that automatic techniques are insufficient, there are several other questions that need to be addressed, such as what is the scope of this annotation task, what other resources, including other annotation layers, might the annotators need access to, and what level of training will they require?

For example, with respect to defining the scope of the task, given a co-reference task, are bridging references and event references to be considered as well? Adding these types of co-reference might make the annotation task more difficult for the annotators, but on the other hand, it might be impossible to define a coherent co-reference task that does not include them.

With respect to other resources, given a semantic or pragmatic target area, the annotators might need access to prior syntactic or semantic annotation. For example, if a researcher is interested in relative clauses, is it possible to annotate a corpus for relative clauses in isolation without having to consider the overall syntactic structure of the sentence? If not, will it matter whether the syntactic analysis is phrase-structure based or dependency based? Alternatively, if the annotation task is sense tagging, which sense inventory will be the most appropriate lexical resource? How fine-grained do the senses need to
be? If a parallel corpus is being annotated with sense tags, does a bi-lingual sense inventory need to be used?

As an example of determining necessary training, if a researcher is interested in providing syntactic analyses of sentences from biomedical texts, do all of the annotators need to become experts in biomedical terminology as well as in syntax?

The answers to any of these questions cannot be determined precisely without considering the task as a whole, the cognitive load on the annotators and the availability of appropriate corpora, resources and tools. We will return to this topic at the end of this section, but for now will assume that a researcher starts with at least a general idea of the phenomena of interest in mind.

3.2 Choosing a target corpus

The criteria for corpus selection depend closely on the objectives for intended use. A large amount of data in a single genre, with as little variation in topic as possible, will yield the best possible performance when tested on similar data. If the characteristics of the test corpus are known in advance, then matching them as closely as possible in the training corpus is effective. However, a significant decrease in performance can be expected when testing on a disparate corpus. The same amount of data selected from a broader, more representative set of documents will yield lower initial performance but more robust results when tested on diverse data. The field is only too familiar with the degradation in performance that occurs when parsers trained on the 1M word Wall Street Journal Treebank are tested on different corpora. This was showcased in the 2005 CoNLL shared task for semantic role labeling (SRL) which included an evaluation on the Brown Corpus, to "test the robustness
of the presented systems” (Carreras & M`arques, 2005). The Charniak POS
tagger degrades by 5%, and the Charniak parser F score degrades by 8%, from
88.25% to 80.84%. For the 19 systems participating in the Semantic Role La-
beling evaluation, there was in general a 10% performance decrease from WSJ
to Brown. The DARPA-GALE funded OntoNotes project (see Section 2.8) is
specifically targeting 200K and 300K selections of data from Broadcast News,
Broadcast Conversation (talk shows), Newsgroups and Weblogs for treebank-
ing, propbanking, sense tagging and coreference annotation. The assumption
is that a more balanced, more representative training corpus will improve the
portability of systems trained on the data. For systems that are intended to
be broad coverage, portability issues are of paramount importance.

**Isolated Sentences**

Alternatively, broader coverage can be achieved by augmenting a target cor-
pus with a hand selected set of constructions. Standard evaluation against
a Treebank simply selects a 10% or smaller chunk of the data for evaluation
purposes. This has the benefit of testing against naturally occurring sentences,
but there is no control over which types of phenomena, such as wh-movement,
gapping, reduced relative clauses, etc., are covered. Unfortunately, a particu-
lar corpus may provide few instances of rare phenomena, and therefore this
type of testing may not adequately cover performance on these types of con-
structions, especially if the initial corpus is small. If necessary a set of selected
instances of rare constructions that are poorly represented in the training and
test corpus can be prepared and added to the Treebank corpora. For example,
(18) is an example of a sentence that has a participial modifier, sometimes
also called a reduced relative, that is in the passive voice and coincides with an
ininitival construction. This may not occur frequently, but a few more similar
examples should provide a statistical parser with sufficient training material to correctly analyze it when it does occur.

(18) “The student, found to be failing several subjects, was still promoted to the next grade.”

Another technique, given a large enough original target corpus, would be to carefully select a subset for annotation that ensures coverage of particular phenomena. A good illustration of hand selection/construction of instances of particular phenomena are the Test Suites for English pioneered by Stephen Oepen and Dan Flickinger (Oepen & Flickinger, 1998; Oepen et al., 2002). The following examples illustrate the types of phenomena with marked word order for which a parser might require additional examples.3

- Heavy NP-shift: “We saw on Tuesday a most amazing film.”
- Relative clause extraposition: “Someone walked in whom I hadn’t met.”
- Locative inversion: “In the corner stood an antique coatrack.”

The difficulty of corpus selection is exacerbated when primarily lexical phenomena are under consideration. Discourse oriented annotation tasks such as coreference or discourse connectives require coherent chunks of text. Unfortunately, even a 1M word corpus of coherent articles such as the WSJ Treebank will not contain sufficient numbers of representative instances of most verbs to be suitable as a training corpus for semantic role labeling or sense tagging. Over half the verbs in WordNet are not present at all, and for those that are present, two thirds (2167 out of 3100) occur less than 10 times. Accurate automatic performance on these verbs can only be achieved by augmenting

---

3 Dan Flickinger provided these examples during an oral presentation at an invitation only Treebank workshop held in conjunction with NAACL07, http://faculty.washington.edu/fxia/treebank/workshop07/formalisms/hpsg.pdf
the WSJ corpus with additional instances. Yet annotating an additional 10M words is clearly not feasible, and would also involve extremely tedious, unnecessary annotation of the 700 verbs that constitute 80% of the token coverage, and already have sufficient instances. The only solution is to once again augment the original target corpus with selected instances from other sources, with the knowledge that the lack of coherence in the augmentation will render it of little use to discourse annotation tasks.

Similar in spirit to the subset selection approach mentioned above, the OntoNotes sense-tagging group at Colorado is currently experimenting with both active learning and language modeling as techniques for finding more examples of rare senses to provide a more even distribution of the training data for a particular lemma (Dligach & Palmer, 2009). The arithmetic sense of *add* is perhaps the most familiar one, but it actually occurs quite infrequently. The *say* sense, as in “And I miss you, too.” *he added mendaciously* occurs by far more frequently. However, in certain domains it would be critical to correctly detect the arithmetic sense, and a few carefully selected additional training examples can significantly improve performance for this type of lemma.

*Parallel Corpora*

Perhaps the greatest challenge to corpus selection involves parallel corpora. Parallel Treebanks and PropBanks are of increasing interest to syntax- and semantics-based statistical machine translation efforts, especially for evaluation purposes.\(^4\) However, the ideal parallel corpus is almost impossible to find. It should be equally fluent with respect to the both the source and target languages, yet at the same time provide a translation that is as literal

\(^4\) The OntoNotes group, at the insistent request of the BBN, IBM and SRI teams, is currently treebanking and propbanking all of the evaluation data from the first three years of the program.
as possible and where each individual sentence can be aligned with a corresponding translated sentence, criteria that are at best at odds with each other. Parliamentary proceedings, such as the Hansards (the official records of the Canadian Parliament); the documents available through Europa (the European Union on-line); and radio shows that are simultaneously broadcast in multiple languages, such as FBIS, offer the most promising sources and are treasured by the community. Indeed, the availability of the Hansards sparked a major transformation in machine translation. These are kept by law in both French and English, and may be legally reproduced and distributed as long as “it is accurately reproduced and that it does not offend the dignity of the House of Commons or one of its Members.” The researchers at IBM were thus provided with a large parallel French/English corpus of closely aligned, literal translations that proved ideal for statistical word alignment techniques (Brown, et. al., 1990). The astonishingly accurate translations their system was able to produce revolutionized the Machine Translation field, and their approach is the basis for increasingly accurate statistical machine translation of Chinese and Arabic to English (funded by DARPA-GALE).

3.3 Annotation efficiency and consistency

When considering which phenomena are to be annotated, it is crucial to think through the decision making process the annotators will be faced with, and whether the task is best done as a single step or with multiple passes. If it is to be done in multiple passes, then the boundaries between the different layers of annotation need to be clearly defined. Sometimes the most time intensive step in the annotation process is comprehending the sentence or phrase to be annotated. There is always a tradeoff between defining simple, modular annotation steps with coherent decision spaces and multiplying the number
of times a sentence has to be comprehended. The success of the annotation process will depend almost entirely on how constrained the choices are and on how clear and comprehensive the guidelines are. It is also important to consider the efficacy of the resulting annotations with respect to training machine learning systems. Although it is no absolute guarantee, human accuracy at the task provides an encouraging indication of at least potential system accuracy. Poor human performance almost certainly presages poor system performance.

**Annotation Errors**

A primary goal in achieving efficient, consistent annotation is reducing unnecessary annotation errors. Apart from expected errors caused by carelessness or fatigue, which can usually be caught through double annotation, other annotation disagreements are almost entirely a direct result of confusion about the guidelines. The annotator may not have read the guidelines thoroughly enough, but more often the guidelines are themselves vague or ambiguous with respect to particular phenomena. It is also sometimes the case that the instances in the data are also vague and/or ambiguous, and open to multiple interpretations. It is important to give the annotators an escape hatch when they do not have a clear intuition about an appropriate label, so that they do not spend too much time agonizing.

**Alternative Guideline Styles**

Each different type of annotation requires a stable, language independent methodology based on guidelines and widely accessible tools. The guidelines need to explicate the details of each individual annotation type as well as interactions between the different types of annotation. For instance, the guidelines for the Proposition Bank outline a process that begins with cre-
Creating a Frameset for each individual verb in the corpus to be annotated. The Framesets provide invaluable additional direction for the annotation of the individual verbs over and above the general annotation guidelines. The NomBank annotation in turn begins by referencing the verb Framesets for associated nominalizations whenever possible. The same approach has been used successfully for the Chinese PropBank/NomBank. In contrast, the guidelines for the Treebank constitute over 300 pages of detailed syntactic description which have to be thoroughly analyzed and memorized before a treebanker can be considered to be fully trained.

Syntactic parsing, or treebanking, is undoubtedly one of the most demanding annotation tasks. Every single possible syntactic phenomenon has to be accounted for in the guidelines, with examples and explanations, and clearly distinguished from other, similar phenomena that it could be confused with, such as subject-control and object-control verbs, raising verbs, etc. The general rule of thumb is that the treebanking guidelines should not be finalized until at least 100K words have been successfully treebanked, which can easily take a year. It takes 6 months to fully train a treebanker, and there are no shortcuts. For phrase structure treebanking, starting with a solid grounding in generative grammar helps, but since the guidelines make many departures from an exact theoretical interpretation, the treebanker also has to be flexible and open minded, and not adhere too rigidly to theoretical generative grammar. However, given highly motivated annotators with a thorough understanding of syntax and the ability to pay close attention to detail, inter-annotator agreement rates of 95% and above have been achieved for English, Chinese and Korean treebanking.
OntoNotes verb sense tagging, on the other hand, requires a very short training period of approximately 20 to 30 hours, which can take as little as 2 weeks. The guidelines amount to only 11 pages, 4 of which are very detailed instructions for logging onto unix and running the annotation tool. Sense taggers only need to know enough syntax to distinguish certain parts-of-speech, such as a main verb as opposed to a verbal past participle used as a modifier, or to recognize the difference between verb arguments that are noun phrases rather than sentential complements. One of the reasons for this brevity is that all of the information for distinguishing between the different senses of a verb has to be made explicit in the entry for that verb in the sense inventory. Each verb is different, and there are no overarching general principles for disambiguation that apply equally to all verbs. The sense inventory itself is the most critical component of the sense tagging guidelines.

The Annotation Process

As an illustration of the contrast between treebanking and sense tagging, treebanking is done on coherent text, sentence by sentence. Having the entire discourse context in mind can be helpful to the treebanker faced with referring pronouns and ambiguous attachment choices. On the other hand, sense tagging is most effectively done using a lexical sample approach, where all of the sentences from the target corpus containing the lemma to be tagged are extracted and displayed to the tagger in a single task. While tagging these instances the tagger only has to have the single lemma and its senses in mind, and does not have to stop and become familiar with several new entries for every sentence to be tagged.\footnote{This benefit has to be weighed against how many times the sentence will be read. If all the content words are to be tagged, and if several of them are polysemous (not likely), the sentence may be re-read several times.} This approach also facilitates the tagger making
consistent choices for that lemma. Even so, if the lemma has too many senses, the cognitive load on the tagger is simply too high. All of the senses cannot be kept in mind, and the tagger has to repeatedly read through the sense entry looking for an appropriate sense label for each instance. This slows down the process and leads to inconsistencies, since the relevant sense might easily be missed. A useful number to keep in mind is Miller’s 7 plus or minus 2. Taggers can manage up to 9 or 10 senses, or even in some cases as many as 12. More than that leads to inconsistency and delay. For the GALE OntoNotes annotation, senses for verb particle constructions for highly polysemous verbs (often accounting for close to half the instances) were split off from the entry for the main verb, and the instances for the verb were tagged in two passes. The first pass allowed for selecting one of 10 to 12 entries for the verb or an additional entry for Multi-word expressions including Verb Particle constructions (MWE). The instances given the MWE tag were tagged in a second pass using a separate entry which split the verb particle constructions as well as idioms and metaphors into several senses. Overall, OntoNotes sense tagging has an ITA of 89% and is 3 to 4 times as fast as WordNet sense tagging.

Determining Tagging Candidates

Determining what constitutes an item to be tagged is just as important as knowing the set of possible labels. In treebanking the span is the entire sentence, so there is no room for confusion. For sense tagging, it is a single lexical item (or a multi-word expression based on a specific item) which has been predetermined and appears highlighted in the sentence. However, the choice of tagging candidates is not always so clear cut. For the ACE Event tagging, the annotators had to select from a sentence the string of words that corresponded to each event participant, a major source of inter-annotator disagreement. For
the initial TimeML annotation, the annotators could pick and choose which
events in a paragraph were supposed to have temporal relations, again, a ma-
jor source of disagreement. A major simplifying assumption for the PropBank
annotation was the decision to base it on the existing Treebank parses. The
events corresponded to clausal verbs, and the event participants that received
semantic role labels were all arguments of the verb. The annotators simply
had to choose a node in the tree that corresponded to a verb argument and
assign a label to it; the span of the node was already pre-determined.

The Necessity of Pilots

Finally, no matter how much thought has gone into defining the guidelines
and constraining annotator choices, a pilot annotation phase is still essential.
Annotators may not interpret the guidelines as intended, the data may present
much more variance than was predicted, there could be (and almost certainly
are) bugs in the annotation tools. Guidelines should never be finalized until
they have been thoroughly tested on a substantial representative data set with
several annotators. Any annotation proposal should include time for a pilot
test, revision of the guidelines, and then another pilot test with additional
revisions. The goal of guidelines development is to define a clear, unambigu-
ous task which can be decided by rapid application of human intuition. If
annotation agreement is lower than expected, it is much more likely the fault
of the guidelines or the task definition than the fault of the annotators.

3.4 Annotation Infrastructure and tools

Because linguistic annotation is generally considered to be a “data” project,
annotation infrastructure and tools, which involve programming support, is an
often overlooked area. It is not uncommon for a linguistic annotation project
to have insufficient funds to hire programmers to provide necessary program-
ning support. This results in annotators having to make do with sub-optimal
annotation tools. Poor tools have a negative impact on both the quality and
the quantity of the annotated data. This section will outline the infrastructure
and tool needs before, during and after annotation. For any large-scale, pro-
duction level annotation projects, having the proper annotation infrastructure
in place at the beginning, as well as equipping the annotator with user-friendly
annotation tools, is essential for the success of the project.

Task Assignment

For the infrastructure of large-scale annotation projects, before human an-
notation can take place, it is essential to think through two key issues: an-
notator management and data flow. Large-scale annotation projects cannot
be done by one person and typically involve multiple annotators. Having a
clear idea of who can access what data is crucial. For sense tagging and prop-
banking, where the standard practice is to perform double-blind annotation
where two (and only two) annotators are asked to annotate the same data, it
is virtually impossible to ask the annotators themselves to keep track of which
data they should or should not annotate. Sense taggers and propbankers are
typically part-time annotators, and they tend to work on an annotation pro-
ject for variable lengths of time. It is often a luxury to expect the same two
annotators to finish annotating the same data during the lifetime of the pro-
ject. As a result, it is necessary to build into the annotation infrastructure
a mechanism to distribute annotation assignments to annotators so that the
annotator can simply take the next available assignment and get on it. Such
a mechanism ensures that a given chunk of data is always double annotated
if that is the goal and that an annotator does not accidentally re-annotate
data that already has double annotations or skip data that needs annotations; mistakes that inevitably happen if the annotators are left to their own devices.

Data Flow Management

In addition to annotator management, it is also important to build into the infrastructure functionalities for data flow management. Annotation is expensive and time-intensive, and it is frustrating to lose annotated data. Annotators are usually linguistic experts who are familiar with the linguistic phenomena they are asked to annotate, but are not necessarily savvy computer users who could be expected to maintain data security. It is necessary to think through how the annotated data should be saved periodically while the annotator is annotating. It is also good practice to maintain version control of the annotated data using a version control facility such as CVS or SVN. Automatic version control systems allows multiple copies of data to be checked in, keeping track of any differences in the most recently checked in versions and of who checked them in and when.  

User Friendly Annotation Tools

After an annotation task is set up and the annotator starts tagging, the annotation tool becomes central to the annotation process. The quality of the annotation tool can have a great impact on annotation efficiency and consistency. Although it seems counter-intuitive, since mouse-based annotation tools have a shorter learning curve and are often preferred by novel annotators, veteran annotators generally prefer keyboard-based annotation interfaces, since

they are faster and are less likely to result in issues such as tendonitis and carpal tunnel syndrome. Long hours spent at annotation have emphasized the ergonomic advantages and greater annotation efficiency of the keyboard. The annotation interface used for the syntactic annotation of the Chinese Treebank is an Emacs-based tool that makes heavy use of keyboard strokes and uses very few mouse clicks. Quite a few mouse-based treebanking tools such as WordFreak and Tred have subsequently been built, and LDC now uses one for English, but the Emacs-based tool is still the preferred treebanking tool for many veteran treebankers, including Chinese and Korean treebankers. It is important to give annotators a choice of keyboard strokes versus mouse clicks.

Annotation tools also have a role in maintaining annotation consistency. If designed properly, an annotation tool can prevent an annotator from entering a label that is not in the tagset associated with the task, or from accidentally deleting or changing the source data, which is usually forbidden. Maintainability, customizability and portability are also important considerations when designing an annotation tool. An annotation tool is often still used long after its original developers have moved on, so someone else needs to maintain it. It is thus advisable that the tool be well documented and written in a widely used programming language. Multilingual annotation is increasingly gaining in popularity, so portability to other natural languages is also an important consideration when choosing a programming language.

Postprocessing

The annotation process does not stop when the human annotator finishes manual annotation. The output of the annotation tool is often not in the final format that the annotation data users expect for processing. Also, for
quality-control purposes, there is often a data validation process after the human annotation is done. For syntactic parsing, this validation can check if the parse is in a valid format, for example, if the left and right brackets match up and if the syntactic labels in the annotated data are all legitimate labels. For sense tagging, the validation process can check if the sense numbers correspond to the sense inventory entry choices. Certain predictable annotation errors can also be automatically detected and corrected. Just as there is user-friendly software, there is also user-friendly data. Not all users of linguistically annotated data are well-versed in the linguistic concepts and their linguistic justifications as encoded in the annotations, and some will lose interest if the representation is too complicated for them to understand. Putting the annotations in an easily understood format can maximize the usability of the data that has taken so much effort to produce.

3.5 Annotation Evaluation

There are two main aspects to evaluating the annotation itself. One has to do with extrinsic measures of the annotation validity and consistency. The other focuses on the agreement between the annotators (ITA), which, since common wisdom treats human annotator agreement figures as an upper bound for system performance, is of central importance (see Chapter 12 for a more detailed discussion).

The question then arises of what to do with annotator disagreements? Are they just mistakes on the part of one annotator that need to be corrected via an adjudication mechanism? This is in fact often the case, but disagreements can also indicate especially vague or ambiguous linguistic phenomena which might need special handling. The OntoNotes sense-tagging project uses ITA as a measure of the clarity of the sense inventory. If ITA is below 90% the
lexicographers are asked to re-examine the groupings of the WordNet senses for that lemma. They cannot examine the actual tagged instances, but they can look at a confusion matrix that shows which senses the annotators disagree on. The confusion matrix sometimes reveals a striking mix-up between two particular senses, pointing the lexicographer to exactly the senses that need clarification. On the other hand, even after a new lexicographer has examined the confusion matrix and the groupings carefully, it may not be at all clear what could or should be changed. Even when all disagreements have been adjudicated to produce the Gold Standard data, system builders often want information about which lemmas, or senses, have been especially difficult to tag, and the original annotator disagreements can be a useful source of this type of information. Evaluation techniques can be weighted to penalize systems less for missing the more difficult cases.

The most straightforward measurement of ITA is simple percentage agreement, and this is also the figure that correlates the most highly with system performance (Chen & Palmer, 2009). However, it has often been pointed out that there is a large discrepancy between 90% ITA when a lemma has a most frequent sense baseline of 85%, and 90% ITA when the most frequent sense baseline is 60%. Chance agreement, and therefore expected agreement, would be much higher with the former than with the latter. The kappa, $k$, coefficient of agreement can take this into account, and is generally considered as providing a more accurate assessment of how much value the annotation is actually adding. The basic technique subtracts expected agreement, $E$, from observed agreement, $A$, and then divides it by 1 minus the expected agreement, as given below.

$$k = \frac{A - E}{1 - E}$$
There are several subtleties in how the expected agreement can be calculated, depending on whether or not there are more than two annotators, what the expected distribution of choices would be, and whether or not annotator bias needs to be taken into account. Artstein and Poesio do an excellent job of surveying the state of the art with respect to these various options (Artstein & Poesio, 2008). In general a kappa score of .8 or higher is considered desirable.

An equally important consideration in measuring ITA is to decide exactly what is being taken into consideration. If treebankers are being evaluated, a simple Parseval score (Black et al., 1991) which matches the sequences of words that have been bracketed together is usually deemed sufficient. This technique produces Precision (of the total number of bracketed constituents produced by an annotator, what percentage are correct) and Recall (what percentage of the correct possible bracketed constituents did the annotator produce) figures as well as the number of crossing brackets. The F-Score is a weighted harmonic mean of Precision and Recall. These scores may or may not take the labels of those bracketed phrases into account. For the Chinese Treebank, based on a randomly selected 20% portion of the corpus, the F-score for the average ITA is 93.8%. After discrepancies between the annotators are reconciled and a Gold Standard produced, Annotator-Gold Standard comparisons, or accuracy comparisons, can be made. For the Chinese Treebank, the F-score for average accuracy was 96.7%.

However, many researchers feel the need for parsing evaluations that are more stringent than Parseval (Carroll et al., 2002, 2003; Hajič et al., 2009), which would translate into more detailed annotator comparisons as well.

With respect to sense tagging, the determination of ITA is from one perspective more straightforward. Senseval (Palmer et al., 2001) and Semeval...
(Pradhan et al., 2007b) evaluations typically provide pointers to the words to be tagged, so recall is always 100% and there is no need to calculate an F-score. Precision is therefore equivalent to accuracy. However, as discussed in Chapter 12, if hierarchical sense entries are provided, the correctness of an answer tag may be weighted depending on its closeness to the correct tag in the hierarchy, making things more complex. In the same way, annotator agreements and disagreements on sense tags can be weighted based on the relatedness of the sense tags chosen.

Coreference annotation presents yet another set of challenges, since coreferences typically consist of sets of mentions. The question is how to score two coreference sets which are almost, but not quite, identical. Artstein and Poesio (Artstein & Poesio, 2008) offer useful insights on this issue.

### 3.6 Pre-processing

There are several questions to be addressed when considering pre-processing:

- What types of pre-processing might facilitate the annotation, and can this be done automatically?
- Does the corpus need to be stripped of headers and extraneous markup? Will they need to be replaced later?
- Is there a pre-existing automatic annotation tool that can be applied without imposing undue bias?

There is considerable evidence that the productivity of manual annotation can be sped up by pre-processing the data with sufficiently accurate automatic taggers (Chiu et al., 2001). This method has been particularly successful with treebanking, where automatic parsers are first run and then the output is hand corrected. Note that this is only useful if the automatic parsers already have
high accuracy. Poor parsers simply slow down the process, and the treebankers
would prefer to start from scratch.\footnote{This is similar to the dismay with which
human translators face the task of hand correcting the output of machine translation systems.} However, in spite of demonstrated productivity gains from automatic pre-processing, current annotation practices
frequently fail to take advantage of this approach, possibly because of the
difficulty of integrating these systems into new annotation tasks.

Even more benefit could be derived from using sophisticated machine
learning techniques to aid in the selection of instances to be tagged, in order
to maximize their utility and minimize the total annotation effort. For simple
classification tasks like Word Sense Disambiguation, there are accepted prac-
tices which utilize automatic WSD systems, such as active learning techniques,
(Chen \textit{et al.}, 2006). However, for more complex annotations such as syntactic
structure, pinpointing novel or unfamiliar items in the data remains a more
challenging problem (Hwa, 2004). Fundamental research is needed to develop
informed sampling techniques for complex annotation that can be integrated
into the annotation effort.
4 Conclusion

This chapter has briefly surveyed several specific annotation layers and reviewed general principles for developing annotation projects. Annotation schemes are likely to be as varied as natural languages, and there are a host of reasons for choosing one annotation tool or evaluation technique over another. However, the principles stated by Krippendorf, (Krippendorf, 2004) for a content analysis annotation scheme are equally applicable to linguistic annotation schemes for natural language processing systems.

- It must employ an exhaustively formulated, clear, and usable coding scheme together with step-by-step instructions on how to use it.
- It must use clearly specified criteria concerning the choice of coders (so that others may use such criteria to reproduce the data).
- It must ensure that the coders that generate the data used to measure reproducibility work independently of each other. [Krippendorf, Chap 11]

We all long for the day when unsupervised and semi-supervised techniques will automatically induce the grammars and sense clusters that drive our natural language processing systems. It may be the case that the more effectively and coherently we annotate the linguistic layers that we currently understand, the sooner that day will come.
References


Brants, Sabine, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, & George Smith (2002), The TIGER treebank, in *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol.


Chiou, Fu-Dong, David Chiang, & Martha Palmer (2001), Facilitating Treebank Annotating Using a Statistical Parser, in Proc. of the Human Language Technology Conference (HLT-2001), San Diego, CA.


Fei Xia, Martha Palmer, Chung-hye Han & Aravind Joshi (2001), Automatically Extracting and Comparing Lexicalized Grammars for Different Languages, in *In the Proceedings of the 17th International Joint conference on Artificial Intelligence (IJCAI-2001)*, Seattle, WA.


Hwa, Rebecca (2004), Sample selection for statistical parsing, *Computational Linguistics* 30(3).


Maamouri, Mahamed & Ann Bies (to appear), The Penn Arabic Treebank, in A. Faraghly & K. Megerdoomian (eds.), *Computational Approaches to Arabic Script-Based Languages: Current Implementation in NLP*, CSLI NLP Series.


Palmer, Martha, Hoa Trang Dang, & Christiane Fellbaum (2007), Making fine-grained and coarse-grained sense distinctions, both manually and automatically, *Journal of Natural Language Engineering* 13(2).


Poesio, Massimo (2004), Discourse Annotation and Semantic Annotation in the GNOME Corpus, in *Proceedings of the ACL Workshop on Discourse Annotation*, Barcelona, Spain.

Poesio, Massimo, Florence Bruneseaux, & Laurent Romary (1999), The MATE Meta-scheme for Coherence in Dialogues in Multiple Languages, in *Proceedings of the ACL Workshop on Standards for Discourse Tagging*, Maryland.


Pradhan, Sameer, Edward Loper, Dmitriy Dligach, & Martha Palmer (2007b), SemEval Task-17: English Lexical Sample, SRL and All Words, in *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, held in conjunction with ACL-2007, Prague, the Czech Republic.


Webber, B. & A. Joshi (1998), Anchoring a lexicalized tree-adjoining grammar for discourse, in ACL/COLING Workshop on Discourse Relations and Discourse Markers, Montreal, Canada.


Yi, Szu-Ting, Edward Loper, & Martha Palmer (2007), Can semantic roles generalize across genres?, in In the Proceedings of the Human Language Technology Confere-