Automatic Predicate Argument Analysis of the Penn TreeBank

Martha Palmer, Joseph Rosenzweig and Scott Cotton
CIS Department, University of Pennsylvania
{mpalmer,josephr,cotton}@linc.cis.upenn.edu

1. INTRODUCTION

One of the primary tasks of Information Extraction is recognizing all of the different guises in which a particular type of event can appear. For instance, a meeting between two dignitaries can be referred to as *A meets B* or *A and B meet*, or *a meeting between A and B took place/was held/opened/convened/finished/dragged on* or *A had/presided over a meeting/conference with B*

There are several different lexical items that can be used to refer to the same type of event, and several different predicate argument patterns that can be used to specify the participants. Correctly identifying the type of the event and the roles of the participants is a critical factor in accurate information extraction. In this paper we refer to the specific subtask of participant role identification as predicate argument tagging. The type of syntactic and semantic information associated with verbs in Levin's Preliminary Classification of English verbs, [Levin,93] can be a useful resource for an automatic predicate argument tagging system. For instance, the *'meet'* class includes the following members, *meet, consult, debate* and *visit*, which can all be used to refer to the meeting event type described above. In addition, the following types of syntactic frames are associated with these verbs:

A *met/visited/debated/consulted* B
A *met/visited/debated/consulted* with B.
A and B *met/visited/debated/consulted*
(with each other).

This type of frame information can be specified at the class level, but there is always a certain amount of verb-specific information that must still be associated with the individual lexical items, such as sense distinctions. For the purposes of this paper we will only be considering sense distinctions based on different predicate argument structures. We begin by giving more information about the Levin classes and then describe the system that automatically labels the arguments in a predicate argument structure. We end by giving the results of evaluating this system versus human annotators performing the same task. Our input to the tagger is the Penn TreeBank [Marcus, 94], so the sentences already have accurate syntactic parses associated with them.

2. LEXICON GUIDELINES

As mentioned above, Levin classes provide the theoretical underpinnings for many of our choices for basic predicate-argument structures [Levin, 93]. Levin verb classes are based on the ability of a verb to occur or not occur in pairs of syntactic frames that are in some sense meaning preserving (diathesis alternations). The distribution of syntactic frames in which a verb can appear determines its class membership. The sets of syntactic frames associated with a particular Levin class are not intended to be arbitrary, and they are supposed to reflect underlying semantic components that constrain allowable arguments. For example, *break* verbs and *cut* verbs are similar in that they can all occur as transitives and in the middle construction, *John broke the window, Glass breaks easily, John cut the bread, This loaf cuts easily*. However, only break verbs can also occur in the simple intransitive, *The window broke, *The bread cut*. Notice that for all of these verbs, the subject of the intransitive, *The window* *broke*, plays the same role as the object of the transitive, *John broke the window*. Our goal is to capture this by using consistent argument labels, in this case Arg1 for the *window* in both sentences. So, for example, *shake* and *rock* would get the following annotation:

The earthquake	shook	the building.
Arg0	REL	Arg1

The walls	shook;
Arg1	REL

the building	rocked.
Arg1	REL

VerbNet In a related project funded by NSF, NSF-IIS98-00658, we are currently constructing a lexicon, VerbNet, that is intended to overcome some of the limitations of WordNet, an on-line lexical database of English, [Miller, 90], by addressing specifically the needs of natural language processing applications. This lexicon exploits the systematic link between syntax and semantics that motivates the Levin classes, and thus provides a clear and regular association between syntactic and semantic properties of verbs and verb classes, [Dang, et al, 98, 00, Kipper, et al. 00]. Specific sets of syntactic configurations and appropriate selectional restrictions on arguments are associated with individual senses. This lexicon gives us a first approximation of sense distinctions that are reflected in varying predicate argument structures. As such these entries provide a suitable foundation for directing consistent predicate-argument labeling of training data.

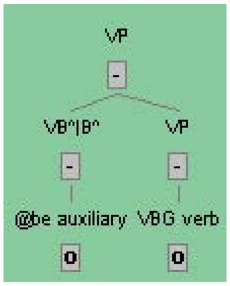
The senses in VerbNet are in turn linked to one or more WordNet senses. Since our focus is predicate-argument structure, we can rely on rigorous and objective sense distinction criteria based on syntax. Purely semantic distinctions, such as those made in WordNet, are subjective and potentially unlimited. Our senses are therefore much more coarse-grained than WordNet, since WordNet senses are purely semantically motivated and often cannot be distinguished syntactically. However, some senses that share syntactic properties can still be distinguished clearly by virtue of different selectional restrictions, which we will also be exploring in the NSF project.

3. AUTOMATIC EXTRACTION OF PREDICATE-ARGUMENT RELATIONS FROM PARSED CORPORA

The predicate-argument analysis of a parse tree from a corpus such as the Treebank corpus is performed in three main phases. First, root forms of inflected words are identified using a morphological analyzer derived from the WordNet stemmer and from inflectional information in machine-readable dictionaries such as the Project Gutenberg version of Webster. Also in this phase, phrasal items such as verb-particle constructions, idioms and compound nominals are identified. An efficient matching algorithm is used which is capable of recognizing both continuous and discontinuous phrases, and phrases where the order of words is not fixed. The matching algorithm makes use of hierarchical declarative constraints on the possible realizations of phrases in the lexicon, and can exploit syntactic contextual cues if a syntactic analysis of the input, such as the parse tree structure of the Treebank, is present. In the next phase, the explicit antecedents of empty constituents are read off from the Treebank annotation, and gaps are filled where implicit linkages have been left unmarked. This is done by heuristic examination of the local syntactic context of traces and relative clause heads. If no explicit markings are present (for automatically generated parses or old-style Treebank parses), they are inferred. Estimated accuracy of this phase of the algorithm is upwards of 90 percent.

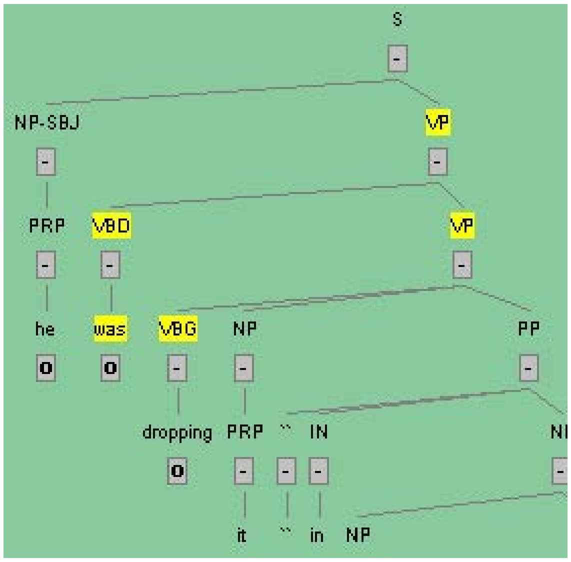
Finally, an efficient tree-template pattern matcher is run on the Treebank parse trees, to identify syntactic relations that signal a predicate-argument relationship between lexical items. The patterns used are fragmentary tree templates similar to the elementary and auxiliary trees of a Tree Adjoining Grammar [XTAG, 95]. Each template typically corresponds to a predication over one or more arguments. There are approximately 200 templates for: transitive, intransitive and ditransitive verbs operating on their subjects, objects and indirect objects; prenominal and predicate adjectives, operating on the nouns they modify; subordinating conjunctions operating on the two clauses that they link; prepositions; determiners; and so on. The templates are organized into a compact network in which shared substructures need to be listed only once, even when they are present in many templates.

Templates are matched even if they are not contiguous in the tree, as long as the intervening material is well-formed. This allows a transitive template for example to match a sentence where there is an intervening auxiliary verb between the subject and the main transitive verb, as in *He was dropping it*. The mechanism for handling such cases resembles the adjunction mechanism in Tree Adjoining Grammar.



Tree grammar template for progressive auxiliary verb, licensing discontinuity in main verb tree

When a template has been identified, it is instantiated with the lexical items that occur in its predicate and argument positions. Each template is associated with one or more annotated template sets, by means of which it is linked to a bundle of thematic or semantic features, and to a class of lexical items that license the template's occurrence with those features. For instance, if the template is an intransitive verb tree, it will be associated both with an unergative feature bundle, indicating that its subject should have the label Arg0, and also with an unaccusative bundle where the subject is marked as Arg1. Which of the feature bundles gets used depends on the semantic class of the word that appears in the predicate position of the template. If the predicate is a causative verb that takes the unaccusative alternation, the subject will be assigned the Arg1 label. If however it is a verb of creation, for example, the subject will be an Arg0. The verb semantics that inform the predicate-argument extractor are theoretically motivated by the Levin classes [Levin, 93], but the actual lexical information it uses is not derived from Levin's work. Rather, it draws on information available in the WordNet 1.6 database [Miller, 90] and on frame codes are derived from the annotation scheme used in the Susanne corpus [Sampson, 95].



Recognition of progressive auxiliary tree which modifies and splits transitive-verb tree for drop in Treebank corpus

For example, one entry for the verb *develop* specifies its WordNet synset membership, and indicates its participation in the unaccusative alternation with the code **o_can_become_s**

develop **SF:so_N_N+W:svJ3W_W:svIM2+o_can_become_s**

The prefix **SF:** signifies that this is a frame code derived from the Susanne corpus. Each frame code picks out a lexical class of the words that take it, and the frame codes are organized into an inheritance network as well. The frame codes in turn are linked to annotated template sets, which describe how these frames can actually appear in the syntactic bracketing format of the TreeBank. In the case of the above frame code for an alternating transitive verb, two template sets are linked: **TG:V_so_N_N** for the frame with a subject and an object (here notated with **s** and **o**); and **TG:V_s_N+causative**, for the unaccusative frame. Each of the template sets lists tree-grammar templates for all the variations of syntactic structure that its corresponding frame may take on. A template for the canonical structure of a simple declarative sentence involving that frame will be present in the set, but additional templates will be added for the forms the frame takes in relative clauses, questions, or passive constructions.

The features for each set are listed separately from the templates, with indications of where they should be interpreted within the various template structures. Hence the template set **TG:V_s_N+causative** includes the feature **TGC:subject+print_as=TGPL:arg1** as part of its feature bundle. This serves to associate the label Arg1 with the **subject** node in each template in the set. When the predicate-argument extractor is able to instantiate such a template, thereby connecting its **subject** node with a piece of a TreeBank tree, it knows to print that piece of the tree as Arg1 of the predicate for that template. If another annotated feature set were active instead, for instance in a case where the predicate of the template does not belong to a verb class which licenses the unaccusative frame code and its associated annotated template set (**TG:V_s_N+causative**), the label of the subject might be different.

4. EVALUATION

The current implementation of the tagger assigns predicate argument structures to all of the 6500 verbs that occur in the Penn Treebank. However, our evaluation of its accuracy is not yet so comprehensive. Our first preliminary evaluation of the performance of the tagger was based on a 5000 word section of the Penn TreeBank. The tagger was run on this, and the argument labeling was subsequently hand corrected by a linguistics graduate student, giving an accuracy rate of 81% out of 160 predicate argument structures. We have since automatically tagged and hand corrected an additional 660 predicate argument structures, with an accuracy rate of 86%, (556 structures), giving us a combined accuracy rate of 83.7%. There are over 100 verbs involved in the evaluation. The number of possible frames for the verbs in the second test ranges from 13 frames to 30, with the typical number being in the teens. Not all of these frames actually appear in the TreeBank data.

These results compare favorably with the results reported by Gildea and Jurafsky of 80.7% on their development set, (76.9% on the test set.) Their data comes from the Framenet project, [Lowe, et al., 97], which has been in existence for several years, and consisted of over 900 verbs out of 1500 words and almost 50,000 sentences. The Framenet project also uses more fine-grained semantic role labels, although it should be possible to map from our Arg0, Arg1 labels to their labels. They used machine learning techniques applied to human annotated data, whereas our tagger does not currently use statistics at all, and is primarily rule-based. Once we have sufficient amounts of data annotated we plan to experiment with hybrid approaches.

5. ACKNOWLEDGEMENTS

We would like to thank Paul Kingsbury and Chris Walker for their annotation efforts, and Aravind Joshi, Mitch Marcus, Hoa Dang and Christiane Fellbaum for their comments on predicate-argument tagging as a task. This work has been funded by DARPA N66001-00-1-8915 and NSF 9800658.

6. REFERENCES

[1] Hoa Trang Dang, Karin Kipper, and Martha Palmer. Integrating compositional semantics into a verb lexicon. In Proceedings of the Eighteenth International Conference on Computational

Linguistics (COLING-2000), Saarbr"ucken, Germany, July-August 2000.

[2] Hoa Trang Dang, Karin Kipper, Martha Palmer, and Joseph Rosenzweig. Investigating regular sense extensions based on intersective levin classes. In Proceedings of Coling-ACL98, Montreal, CA, August 1998.

[3] Daniel Gildea and Daniel Jurafsky, Automatic Labeling of Semantic Roles, In Proceedings of the Association for Computational Linguistics Conference, Hong Kong, October, 2000.

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

[5] Beth Levin. English Verb Classes and Alternations A Preliminary Investigation. 1993.

[6] J.B. Lowe, C.F. Baker, and C.J. Fillmore. A frame-semantic approach to semantic annotation. In Proceedings 1997 Siglex Workshop/ANLP97, Washington, D.C., 1997.

[7] Mitch Marcus. The penn treebank: A revised corpus design for extracting predicate argument structure. In Proceedings of the ARPA Human Language Technology Workshop, Princeton, NJ, March 1994.

[8] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Five papers on wordnet. Technical Report 43, Cognitive Science Laboratory, Princeton University, July 1990.

[9] Martha Palmer, Hoa Trang Dang, and Joseph Rosenzweig. Sense tagging the penn treebank. In Proceedings of the Second Language Resources and Evaluation Conference, Athens, Greece.

[10] The XTAG-Group. A Lexicalized Tree Adjoining Grammar for English. Technical Report IRCS 95-03, University of Pennsylvania, 1995.