XARA: An XML- and rule-based semantic role labeler

Gerwert Stevens
University of Utrecht, the Netherlands
gerwert.stevens@let.uu.nl

Abstract

XARA is a rule-based PropBank labeler for Alpino XML files, written in Java. I used XARA in my research on semantic role labeling in a Dutch corpus to bootstrap a dependency treebank with semantic roles. Rules in XARA are based on XPath expressions, which makes it a versatile tool that is applicable to other treebanks as well.

In addition to automatic role annotation, XARA is able to extract training instances (sets of features) from an XML based treebank. Such an instance base can be used to train machine learning algorithms for automatic semantic role labeling (SRL). In my semantic role labeling research, I used the Tilburg Memory Learner (TiMBL) for this purpose.

1 Introduction

Ever since the pioneering article of Gildea and Jurafsky (2002), there has been an increasing interest in automatic semantic role labeling (SRL). In general, classification algorithms (a supervised machine learning strategy) are used for this purpose. Manual annotated corpora provide a gold standard for such classifiers.

Starting manual annotation from scratch is very time consuming and therefore expensive. A possible solution is to start from a (partially) automatically annotated corpus. In fact, this reduces the manual *annotation* task to a manual *correction* task. Initial automatic annotation of a corpus is often referred to as *bootstrapping* or *unsupervised SRL*.

In recent years relatively little effort has gone into the development of unsupervised SRL systems. This is partly because semantically annotated English corpora, such as PropBank (Kingsbury et al., 2002) and FrameNet (Johnson et al., 2002), currently contain enough data to develop and test SRL systems based on machine learning. Therefore, bootstrapping large collections of English texts has no priority anymore. For languages other than English however, annotated corpora are rare and still very much needed. Therefore, the development of bootstrapping techniques is very relevant.

One of the languages for which the creation of semantically annotated corpora has lagged dramatically behind, is Dutch. Within the project *Dutch Language Corpus Initiative* (D-Coi)[1], the first steps have been taken towards the development of a large semantically annotated Dutch corpus. The D-Coi project is a preparatory project which will deliver a blueprint and the tools needed for the construction of a 500-million-word reference corpus of contemporary written Dutch. The corpus will be annotated with several layers of annotation, amongst others with semantic roles.

In the context of this project, I developed XARA: (**X**ML-based **A**utomatic **R**ole-labeler for **A**lpino-trees). In my research, XARA was used for two purposes:

- Bootstrap a dependency treebank with semantic roles

[1] http://lands.let.ru.nl/projects/d-coi/

Proceedings of the Linguistic Annotation Workshop, pages 113–116,
Prague, June 2007.

- Extract an instance base for the training of a semantic role classifier.

2 Rule-based role labeling

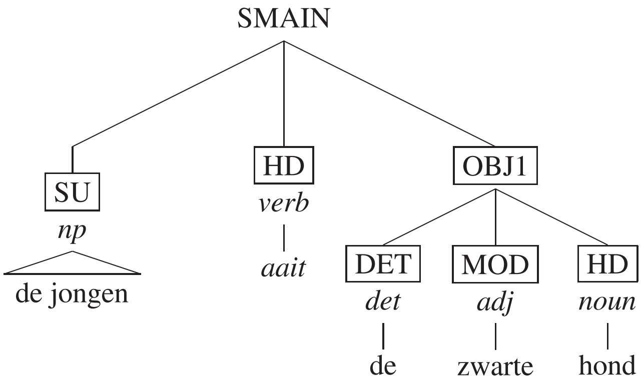
2.1 The Alpino XML-format

The input for the semantic role tagger is a set of sentences annotated by the Dutch dependency parser Alpino (Bouma et al., 2000) [2]. Alpino is based on a hand-crafted Head-driven Phrase Structure Grammar (HPSG).

The annotation scheme of Alpino dependency trees is based on the Spoken Dutch Corpus (CGN) (Oostdijk, 2002) annotation format. In Alpino trees the same labels are used as in their CGN counterparts and nodes are structured in the same way. The XML-format used to store dependency trees however differs. In the CGN, sentences are stored in the TIGER-XML format (Lezius, 2002) [3], Alpino uses its own XML format to store parsed sentences (Bouma and Kloosterman, 2002). In our treebank, every sentence was encoded in a separate XML file. An example of an Alpino dependency tree annotated with semantic roles is shown in figure 1. Below, the corresponding XML output is shown:

```
<node rel="top">
 <node cat="top" rel="top">
  <node cat="smain" rel="--">
  <node cat="np" rel="su">
   <node pos="det" rel="det" word="de"/>
   <node pos="noun" rel="hd" word="jongen"/>
  </node>
  <node pos="verb" rel="hd" word="aait"/>
  <node cat="np" rel="obj1">
   <node pos="det" rel="det" word="de"/>
   <node pos="adj" rel="mod" word="zwarte"/>
   <node pos="noun" rel="hd" word="hond"/>
  </node>
 </node>
</node>
```

Figure 1: Example CGN dependency graph ('The boy pets the black dog')



The structure of Alpino XML documents directly corresponds to the structure of the dependency tree: dependency nodes are represented by NODE elements, attributes of the node elements are the c-label, d-label, pos-tag, etc. The format is designed to support a range of linguistic queries on the dependency trees in XPath directly (Bouma and Kloosterman, 2002). XPath (Clark and DeRose, 1999) is a powerful query language for the XML format and it is the cornerstone of XARA's rule-based approach.

I would like to stress that although our SRL research focused on Alpino structures, XARA can be used with any XML-based treebank, thanks to the fact that XPath and XML are widely accepted standards. This property satisfies one of the major design criteria of the system: reusability.

2.2 The annotation process

The input for the tagger is set of directories containing Alpino XML files, called a *treebank*. Each sentence is annotated separately by applying a set of rules. Rules are applied to local dependency domains (subtrees of the complete dependency tree). The local dependency domain to which a rule is applied, is called the rule's *context*. A context is simply defined by an XPath expression which selects a group of nodes.

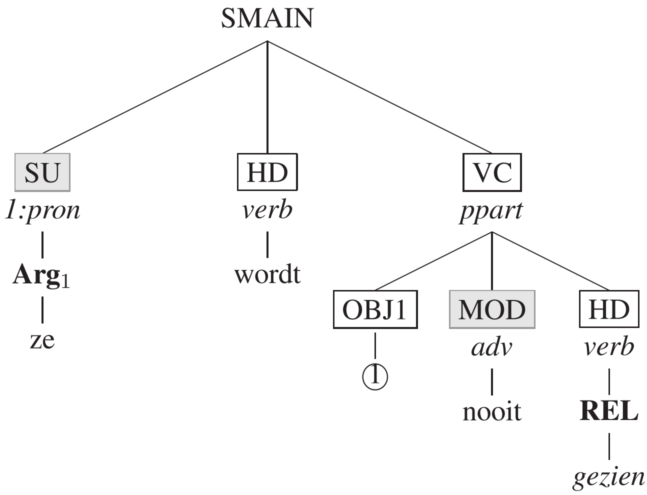
Suppose for example that we want to apply a certain rule to nodes that are part of a passive participle, i.e the context of our rule are passive participles. Passive participles in Alpino trees are local dependency domains with a root node with c-label PPART. An example is shown in figure 2.

The dark colored nodes are the ones we are interested in. To select these nodes, the following XPath expression can be used:

[2] A demonstration of the Alpino parser can be found on the following website: http://ziu.let.rug.nl/vannoord_bin/alpino

[3] see also http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/index.shtml

Figure 2: Example PropBank annotation on a Dependency tree ('She is never seen')



```
//node[@cat='ppart']
[preceding-sibling::
node[@rel='hd' and (@root='word')]]
```

which says that we are looking for nodes with the c-label PPART and the auxiliary verb indicating passive tense (*word*) as preceding sibling.

Once a context is defined, rules can be applied to nodes in this context. Rules consist of an XPath expression which specifies a relative path from the context's root node to the target node and an output label. Upon application of the rule, the target node will be labeled with output label.

The output label can have three kinds of values:

- A positive number n, to label a node with ARG$_n$.
- The value -1, to label the node with the first available numbered argument.
- A string value, to label the node with an arbitrary label, for example an ARGM.

Notice that because the label can be specified as a string value, the set of possible labels is not restricted. In my work, I used PropBank labels, but other labels - such as generic thematic roles - can be used just as well.

Formally, a rule in XARA can be defined as a $(path, label)$ pair. Suppose for example that we want to select direct object nodes in the previously defined context and assign them the label ARG1. This can be formulated as:

```
(./node[@rel='obj1'],1)
```

The first element of this pair is an XPath expression that selects direct object daughters, the second element is a number that specifies which label we want to assign to these target nodes. In this case the label is a positive integer 1, which means the target node will receive the label ARG1. Upon application of a rule, an attribute (”pb”) is added to the target node element in the XML file. This attribute contains the PropBank label.

3 Feature extraction

Besides bootstrapping an unannotated corpus, training a SRL classifier was another important part of my automatic SRL strategy. The learning tool I used for this purpose was TiMBL (Tilburg Memory Based Learner) (Daelemans et al., 2004).

In order to be able to train a TiMBL classifier, a file with training data is needed. Training data is represented as a text file containing instances. Each line in the text file represents a single instance. An instance consists of a set of features separated by commas and a target class. XARA is able to create such an instance base from a set of XML files automatically.

3.1 The automatic feature extraction process

The target instance base consists of predicate/argument pairs encoded in training instances. Each instance contains features of a predicate and its candidate argument. Candidate arguments are nodes (constituents) in the dependency tree. This pair-wise approach is analogous to earlier work by van den Bosch et al. (2004) and Tjong Kim Sang et al. (2005) in which instances were built from verb/phrase pairs from which the phrase parent is an ancestor of the verb.

Once it is clear how instances will be encoded, an instance base can be extracted from the annotated corpus. For example, the following instances can be extracted from the tree in figure 2:

```
zie,passive,mod,adv,#
zie,passive,su,pron,ARG1
```

These two example instances consist of 4 features and a target class each. In this example, the predicate lemma (stem) and voice, and the candidate argument c-label, d-label are used. For null values the hash symbol (#) is specified. The first instance represents the predicate/argument pair $(zie, nooit)$ ('see,never'), the second instance represents the pair (zie, ze) ('see, she').

The extraction of instances from the annotated corpus can be done fully automatically by XARA from the command line. The resulting feature base can be directly used in training a TiMBL classifier.

4 Performance

In order to evaluate the labeling of XARA, the output of XARA's semantic role tagger was compared with the manual corrected annotation of 2,395 sentences. The results are shown in table 1.

Table 1: Overall performance

Precision	**Recall**	**F$_{\beta=1}$**
65,11%	45,83%	53,80

Since current rules in XARA cover only a subset of PropBank labels, recall is notably lower than precision. However, current overall performance of XARA is encouraging. Our expectation is that, especially if the current rule set is improved and/or extended, XARA can be a very useful tool in current and future SRL research.

References

G. Bouma and G. Kloosterman. 2002. Querying dependency treebanks in xml. In *Proceedings of the Third international conference on Language Resources and Evaluation (LREC)*. Gran Canaria.

G. Bouma, G. van Noord, and R. Malouf. 2000. Alpino: wide-coverage computational analysis of dutch.

J. Clark and S. DeRose. 1999. Xml path language (xpath). *W3C Recommendation 16 November 1999*. URL: http://www.w3.org/TR/xpath.

D. Daelemans, D. Zavrel, K. van der Sloot, and A. van den Bosch. 2004. Timbl: Tilburg memory based learner, version 5.1, reference guide. ILK Technical Report Series 04-02, Tilburg University.

D. Gildea and D. Jurafsky. 2002. Automatic labeling of semantic roles. *Comput. Linguist.*, 28(3):245–288.

C. R. Johnson, C. J. Fillmore, M. R. L. Petruck, C. F. Baker, M. J. Ellsworth, J. Ruppenhofer, and E. J. Wood. 2002. *FrameNet:Theory and Practice*.

P. Kingsbury, M. Palmer, and M. Marcus. 2002. Adding semantic annotation to the penn treebank. In *Proceedings of the Human Language Technology Conference (HLT'02)*.

W Lezius. 2002. *Ein Suchwerkzeug für syntaktisch annotierte Textkorpora*. Ph.D. thesis, University of Stuttgart.

N. Oostdijk. 2002. The design of the spoken dutch corpus. In P. Peters, P. Collins, and A. Smith, editors, *New Frontiers of Corpus Research*, pages 105–112. Amsterdam: Rodopi.

E. Tjong Kim Sang, S. Canisius, A. van den Bosch, and T. Bogers. 2005. Applying spelling error correction techniques for improving semantic role labeling. In *Proceedings of the Ninth Conference on Natural Language Learning (CoNLL-2005)*. Ann Arbor, MI, USA.

A. van den Bosch, S. Canisius, W. Daelemans, I. Hendrickx, and E. Tjong Kim Sang. 2004. Memory-based semantic role labeling: Optimizing features, algorithm, and output. In H.T. Ng and E. Riloff, editors, *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*. Boston, MA, USA.