Introduction to Grammar Engineering using HPSG

Master Course, Wintersemester 2005/06
Computational Linguistics and Phonetics, Universität des Saarlandes
Dan Flickinger – dan@coli.uni-sb.de – Office C 7.2:3.06 – Tel. 4118

Course web page: http://lingo.stanford.edu/dan/courses/ge05/
More lexical subcategorization types

Controlled VP complements
(1) The dog pretended to chase the cat.
(2) The dog persuaded the cat to chase the aardvark.
(3) The dog promised to chase the cat.

Sentential complements
(1) The dog hopes the aardvark chases the cat.
(2) The dog knows that the cat chases the aardvark.
(3) The cat wonders whether the dog chases the aardvark.
Cross-categorial generalizations

Adjective and nouns can also select for verbal complements:

1. *The dog was eager to bark.*
2. *The cat was glad the dog chased the aardvark.*
3. *The dog’s attempt to bark bothered the cat.*
4. *The fact that the dog barked bothered the cat.*
Some more interesting types of subcategorization

<table>
<thead>
<tr>
<th>Type</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjective + VP/NP complement</td>
<td>(1) <em>That cat was difficult to chase.</em></td>
</tr>
<tr>
<td></td>
<td>(2) <em>The cat is easy to give to the aardvark.</em></td>
</tr>
<tr>
<td>Verb + particle</td>
<td>(1) <em>That cat looked the answer up.</em></td>
</tr>
<tr>
<td></td>
<td>(2) <em>That cat looked up the answer.</em></td>
</tr>
<tr>
<td></td>
<td>(3) <em>It turned out that the dog barked.</em></td>
</tr>
</tbody>
</table>
Coordination

Verb + particle

(1) *The dog and the cat chased the aardvark.*
(2) *The dog chased the cat and barked.*
(3) *The dog that those cats chased gave that cat and the aardvark to that dog.*
Summary of what you’ve learned in this course

- Notion of grammars for natural languages
- Linguistic theory as framework for analyses
- Formalism for encoding linguistic constraints
- Development environment
- Grammar engineering practices
Properties of a good grammar

- Precision: minimal spurious ambiguity, bidirectionality
- Compositional semantics: generation, applications, evaluation
- Linguistic clarity: minimal redundancy
- Scalability for coverage
- Efficiency for processing
- Documentation: more is better
Linguistic theory: HPSG

- Language-independent hypotheses
  Architecture of sign (lexeme, word, phrase)
  Abstract rule types
  Principles of feature propagation
  Syntax-semantics interface
- Specialization for individual languages
  Sub-regularities
  Idiosyncratic exceptions
Grammar formalism

- Typed feature structures in TDL
  - Multiple inheritance type hierarchy
  - Feature structures with type inference, re-entrancy
  - Basic machinery, including difference lists!
- Language independence
- Separable morphology component
The LKB development environment

- Emacs!
- Linking up text files with loaded grammar
- Interpretation of error messages
- Viewing tools
  - Type hierarchy
  - Parse chart
  - Parse trees
  - MRS semantics
- Batch testing
- Generator
Grammar engineering practices

- Organization of definitions into files
- Analysis of data, construction of hypotheses
- Collaborative work: starting from an existing grammar
- Mastery of the Change/Test/Debug/Test/Celebrate cycle
- Documentation of additions, changes
- Frequent batch testing