Enju Parser

1. BASIC INFORMATION

Tool name

Enju Parser

Overview and purpose of the tool

Enju is a syntactic parser for English. The grammar used by the parser is based on Head Driven Phrase Structure Grammar (HPSG). Enju can analyse syntactic/semantic structures of English sentences can output phrase structure and predicate-argument structures.

A short description of the algorithm

The grammar of Enju is based on the theory of Head-driven Phrase Structure Grammar (HPSG). In HPSG, constraints on the structure of a language are represented with *typed feature structures*. Enju uses a wide-coverage probabilistic HPSG grammar (Miyao & Tsujii, 2002; Miyao & Tsujii 2003; Miyao et al., 2004; Miyao & Tsujii, 2005; Ninomiya et al., 2006; Ninomiya et al., 2007; Miyao & Tsujii, 2008) and an efficient parsing algorithm (Tsuruoka et al., 2003; Ninomiya et al, 2005; Ninomiya et al, 2006; Matzuzaki et al., 2007)

One of the characteristics of HPSG is that most of the constraints on syntax and semantics are represented in lexical entries, while only a small number of grammar rules (corresponding to CFG rules) are defined and they represent general constraints irrelevant to specific words. This is because the constraints on the structure of a sentence are mostly introduced by words.

Syntactic/semantic constraints of words/phrases are represented in the data structure called *sign*. In the current implementation of Enju, the structure of the sign basically follows Pollard and Sag (1994) and LinGO English Resource Grammar (ERG), while the type hierarchy is much simplified and modified not to use complex constraints nor Minimal Recursion Semantics (MRS).

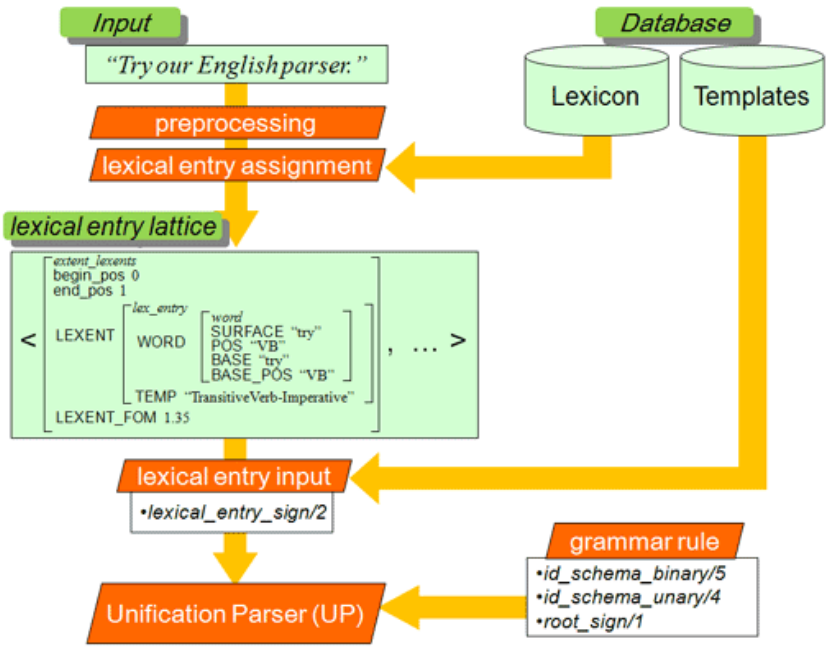
Constraints of phrases include various syntactic features (part-of-speech, agreement, tense, etc.).

The CONT feature contains the predicate-argument structure of the phrase. Predicate-argument structures represent relations of logical subject/object and modifying relations. The CONT feature of the sign of the top node shows the predicate-argument structure of the whole sentence.

PHON: A sequence of words governed by the phrase

SYNSEM
- LOCAL
  - CAT
    - HEAD
      - Constraints inherited from the head daughter
      - MOD: Constraints of a modifying phrase
      - POSTHEAD: Whether this phrase modifies a preceding phrase
    - VAL
      - Subcategorization frame
      - SUBJ: Constraints of left phrases that will be subcategorized
      - COMPS: Constraints of right phrases that will be subcategorized
      - SPR: Constraints of specifiers that will be subcategorized
      - SPEC: Constraints of a specifying phrase
      - CONJ: Constraints of conjuncts
  - CONT: Predicate-argument structure
  - CONX: *Currently not used*
- NONLOCAL: Constraints of long-distance dependency
  - INHER
    - Constraints inherited from the daughters
    - QUE: Constraints of wh-question words
    - REL: Constraints of an antecedent phrase of a relative clause
    - SLASH: Constraints of a phrase in long-distance dependencies
    - F_REL: Constraints of free relatives
  - TO_BIND
    - Constraints bound in this phrase
    - QUE: Constraints of wh-question words
    - REL: Constraints of an antecedent phrase of a relative clause
    - SLASH: Constraints of a phrase in long-distance dependencies
    - F_REL: Constraints of free relatives

The Enju system uses *UP* (http://www.nactem.ac.uk/enju/mayz-manual/up.html, included in the MAYZ package), a general-purpose parser for unification grammars. UP parses a sentence with provided lexical entries and grammar rules. Enju creates the data passed to UP in the following way.



2. TECHNICAL INFORMATION

Software dependencies and system requirements

Different versions of Enju are available that run on Windows, Mac and a variety of Unix operating systems. The latest version of Enju (2.4.2) can run in the following environments:

- Ubuntu 8 for i386 (and many other 32 bit linux distributions)
- CentOS 5.5 for i386
- CentOS 5.5 for x86_64 (and many other 64 bit linux distributions)
- Mac OS X
- Windows (32 bit)

Installation

On Linux or Mac OS

1. Download the latest package for your particular platform (enju-X.Y-PLATFORM.tar.gz) from http://www.nactem.ac.uk/enju/#download

2. Untar the archive into a directory where you would like to install Enju.

```
> tar xvzf enju-X.Y-PLATFORM.tar.gz
```

3. Run `enju-X.Y/enju` to invoke enju.

On Windows

1. Download the latest package for Windows (enju-X.Y-win32.zip) from http://www.nactem.ac.uk/enju/#download

2. Unzip the archive into a directory where you would like to install Enju.

3. Run `enju-win/enju.bat` to invoke enju.

Online demo

Enju can also be tested by using the online demo, available here:

http://www.nactem.ac.uk/enju/demo.html

Execution instructions

To parse sentences, put a file (having **one sentence per line**) to the standard input. For example, when you have the file "RAWTEXT" that contains:

```
He runs the company.
The company that he runs is small.
```

Run the following command.

```
> enju < RAWTEXT > RESULTS
```

Parsing results are output to the file "RESULTS".

You can alternatively use a high-speed parser by using the command "mogura"

```
> mogura < RAWTEXT > RESULTS
```

These commands work in mostly the same way.

When you want to parse texts already tagged with Penn Treebank-style POS tags,

```
> enju -nt < TAGGEDTEXT > RESULTS
```

The default output of the parser is a set of predicate-argument relations. Alternatively, you can get both the phrase structures and predicate-argument relations either in a quasi-XML format or in a stand-off format.

```
> enju -xml < RAWTEXT > RESULTS
> enju -so < RAWTEXT > RESULTS
```

You can also use Enju as a CGI server.

```
> enju -cgi PORT_NUMBER
```

You can access to the port PORT_NUMBER with a CGI query, and receive parsing results in the XML format.

```
http://localhost:PORT_NUMBER/cgililfes/enju?sentence=he+runs+the+company
```

Input/Output data formats

Input data formats

The input is plain text, with one sentence per line.

Output data format

The default output of the parser is a set of predicate-argument relations, so the user can easily acquire semantic relations among words in an input sentence without the burden of analyzing its deep-syntactic structure. Further information about the output format can be found here: http://www.nactem.ac.uk/enju/enju-manual/enju-output-spec.html

Enju can also output both phrase structures and predicate-argument structures in a quasi-XML format. Further information about the XML format can be found here: http://www.nactem.ac.uk/enju/enju-manual/enju-xml-format.html

Integration with external tools

N/A

3. CONTENT INFORMATION

Parsing examples are shown below. Each line in the output represents a predicate-argument relation between two words. For instance, the second line in the first example indicates that there is an "ARG1 (logical subject)" relation between the predicate "run" and the argument "he". Note that the same semantic relations holding among the three words, "he", "run", and "company", are obtained from sentences with differing syntactic structures.

Sentence 1: He runs the company.

ROOT	ROOT	ROOT	ROOT	-1	ROOT	ROOT	runs	run	VBZ	VB	1
runs	run	VBZ	VB	1	verb_arg12	ARG1	He	he	PRP	PRP	0
runs	run	VBZ	VB	1	verb_arg12	ARG2	company	company	NN	NN	3
the	the	DT	DT	2	det_arg1	ARG1	company	company	NN	NN	3

Sentence 2: The company that he runs is small.

ROOT	ROOT	ROOT	ROOT	-1	ROOT	ROOT	is	be	VBZ	VB	5
is	be	VBZ	VB	5	verb_arg12	ARG1	company	company	NN	NN	1
is	be	VBZ	VB	5	verb_arg12	ARG2	small	small	JJ	JJ	6
small	small	JJ	JJ	6	adj_arg1	ARG1	company	company	NN	NN	1
The	the	DT	DT	0	det_arg1	ARG1	company	company	NN	NN	1

that	that	IN	IN	2	relative_arg1	ARG1	company	company	NN	NN	1
runs	run	VBZ	VB	4	verb_arg12	ARG1	he	he	PRP	PRP	3
runs	run	VBZ	VB	4	verb_arg12	ARG2	company	company	NN	NN	1

Running the tool on the 4 KB text on a single core machine with 8 GB RAM takes around 30 milliseconds.

3. LICENCES

The Enju parser is licensed using a proprietary non-exclusive academic use licence. Please see ENJU-LICENCE.txt in the licences directory. Please contact us using the details below if you require a commercial licence.

4. ADMINISTRATIVE INFORMATION

Contact

For further information, please contact Sophia Ananiadou:
sophia.ananiadou@manchester.ac.uk

5. REFERENCES

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. (2007). Efficient HPSG Parsing with Supertagging and CFG-filtering. In *Proceedings of IJCAI 2007*.

Yusuke Miyao and Jun'ichi Tsujii. (2002). Maximum Entropy Estimation for Feature Forests. In *Proceedings of HLT 2002*.

Yusuke Miyao and Jun'ichi Tsujii. (2003). Probabilistic modeling of argument structures including non-local dependencies. In *Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP) 2003*, pp. 285-291

Yusuke Miyao, Takashi Ninomiya, and Jun'ichi Tsujii. (2004). Corpus-oriented Grammar Development for Acquiring a Head-driven Phrase Structure Grammar from the Penn Treebank. In *Proceedings of IJCNLP-04*.

Yusuke Miyao and Jun'ichi Tsujii. (2005). Probabilistic Disambiguation Models for Wide-Coverage HPSG Parsing. In *Proceedings of ACL-2005*, pp. 83-90.

Yusuke Miyao and Jun'ichi Tsujii. (2008). Feature Forest Models for Probabilistic HPSG Parsing. *Computational Linguistics*. 34(1):35--80, MIT Press

Takashi Ninomiya, Yoshimasa Tsuruoka, Yusuke Miyao, and Jun'ichi Tsujii. (2005). Efficacy of Beam Thresholding, Unification Filtering and Hybrid Parsing in Probabilistic HPSG Parsing . In *Proceedings of IWPT 2005*.

Takashi Ninomiya, Takuya Matsuzaki, Yoshimasa Tsuruoka, Yusuke Miyao and Jun'ichi Tsujii. (2006). Extremely Lexicalized Models for Accurate and Fast HPSG Parsing. In *Proceedings of EMNLP 2006*.

Takashi Ninomiya, Yoshimasa Tsuruoka, Yusuke Miyao, Kenjiro Taura and Jun'ichi Tsujii. (2006). Fast and Scalable HPSG Parsing. *Traitement automatique des langues (TAL)*. 46(2). Association pour le Traitement Automatique des Langues.

Takashi Ninomiya, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. (2007). A log-linear model with an n-gram reference distribution for accurate HPSG parsing. In *Proceedings of IWPT 2007*.

C. Pollard and I. A. Sag. (1994). *Head-Driven Phrase Structure Grammar*. University of Chicago Press

Yoshimasa Tsuruoka, Yusuke Miyao, and Jun'ichi Tsujii. (2003). Towards efficient probabilistic HPSG parsing: integrating semantic and syntactic preference to guide the parsing. In *Proceedings of IJCNLP-04 Workshop: Beyond shallow analyses - Formalisms and statistical modeling for deep analyses.*