UIMA/U-Compare Enju Parser

1. BASIC INFORMATION

Tool name

UIMA/U-Compare Enju Parser

Overview and purpose of the tool

Syntactic parser for English. Outputs predicate-argument structures. Also outputs base forms for each token.

The tool is provided as a UIMA[1] (Ferrucci et al., 2006) component, which forms part of the in-built library of components provided with the U-Compare platform (Kano et al., 2009; Kano et al., 2011; see separate META-SHARE record)[2] for building and evaluating text mining workflows. The U-Compare Workbench (see separate META-SHARE record) provides a graphical drag-and drop interface for the rapid creation of workflows.

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).

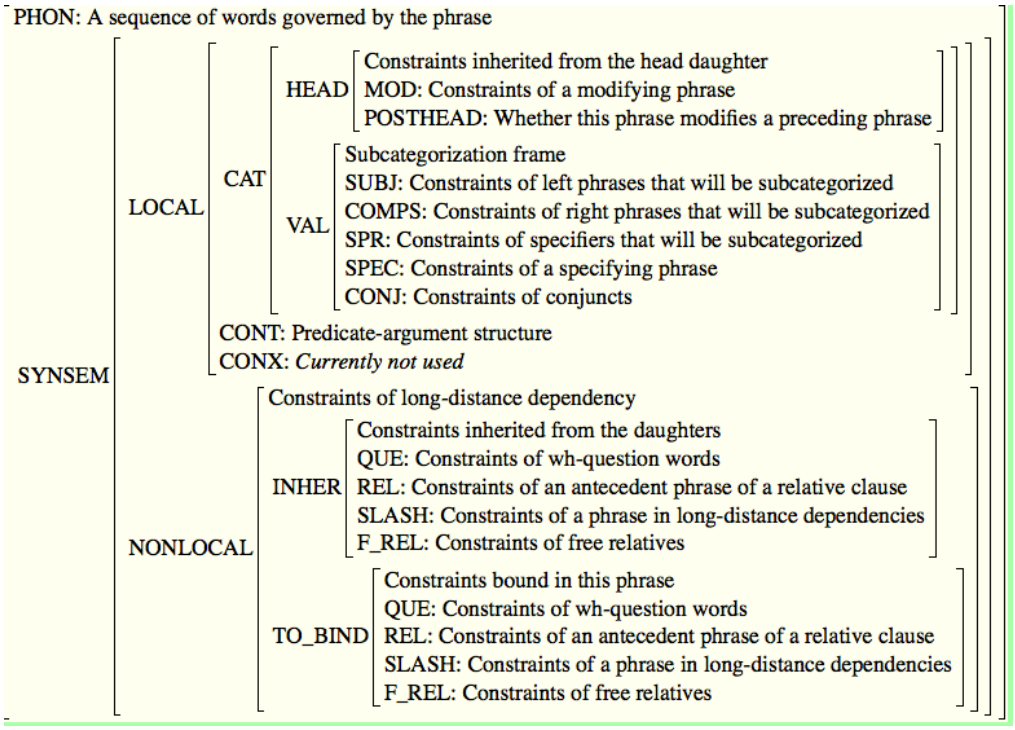
Constraints of phrases include various syntactic features (part-of-speech, agreement, tense, etc.).

---

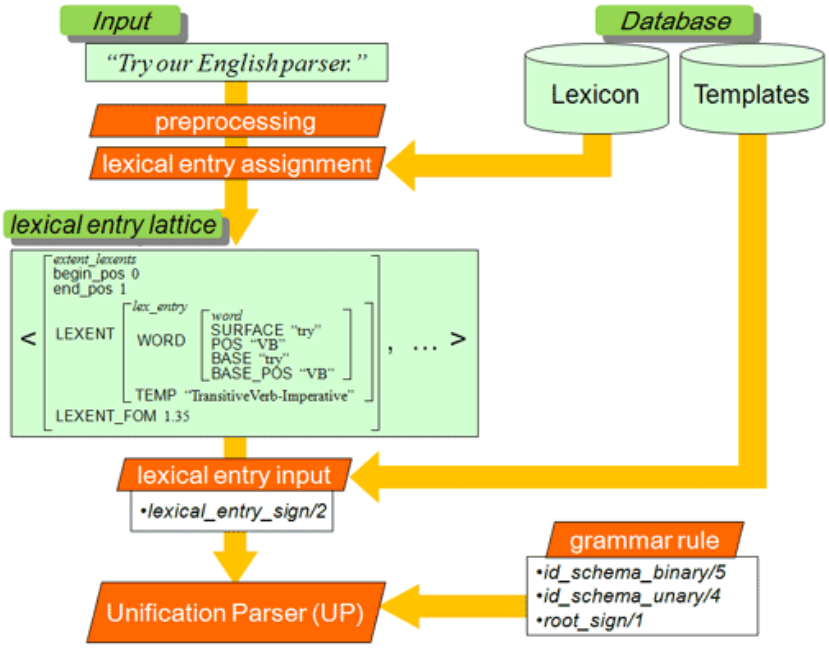
[1] http://uima.apache.org/

[2] http://nactem.ac.uk/ucompare/

The CONT feature represents 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.



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

The tool is provided as a UIMA component wrapped around a web service. Thus, the tool must be run within the Apache UIMA framework. Alternatively, it can be run within the U-Compare framework. The component has been specifically designed to work in U-Compare workflows and is compliant with the U-Compare type system.

Installation

The tool is provided as an in-built component of the U-Compare workbench. However, it can also be used in other UIMA workflows. Since it is packaged as a UIMA component, no specific installation is required, following installation of the UIMA framework and/or U-Compare.

Execution instructions

The tool can be used within U-Compare simply be dragging and dropping it into a workflow using the graphical user interface of the U-Compare workbench. Alternatively, it can be incorporated into other UIMA-based workflows, by following the documentation on the Apache UIMA site. Given that the UIMA component is implemented in Java, the tool is platform-independent.

Input/Output data formats

Input data formats

The input is plain text document that has previously been read into the UIMA Common Analysis Structure (CAS) via a UIMA collection reader component. As a

prerequite, the CAS must contain sentence annotations and POSToken annotations (i.e., token annotations with part of speech information attached). Thus, appropriate components must be executed in the workflow to add these annotations prior to running the Enju parser component.

Output data format

The tool creates Enju Sentence, EnjuToken and EnjuConstituent annotations. The EnjuConstituent annotations encode the predicate-argument relations output by the parser/

Integration with external tools

As mentioned above, the tool can only be run within the UIMA or U-Compare frameworks.

3. CONTENT INFORMATION

Figure 1 shows the output of the tool in the U-Compare workbench. The arcs between words represent the dependency relations in the first sentence. The sample out is taken from the PubMed website (http://www.ncbi.nlm.nih.gov/pubmed/23172825)

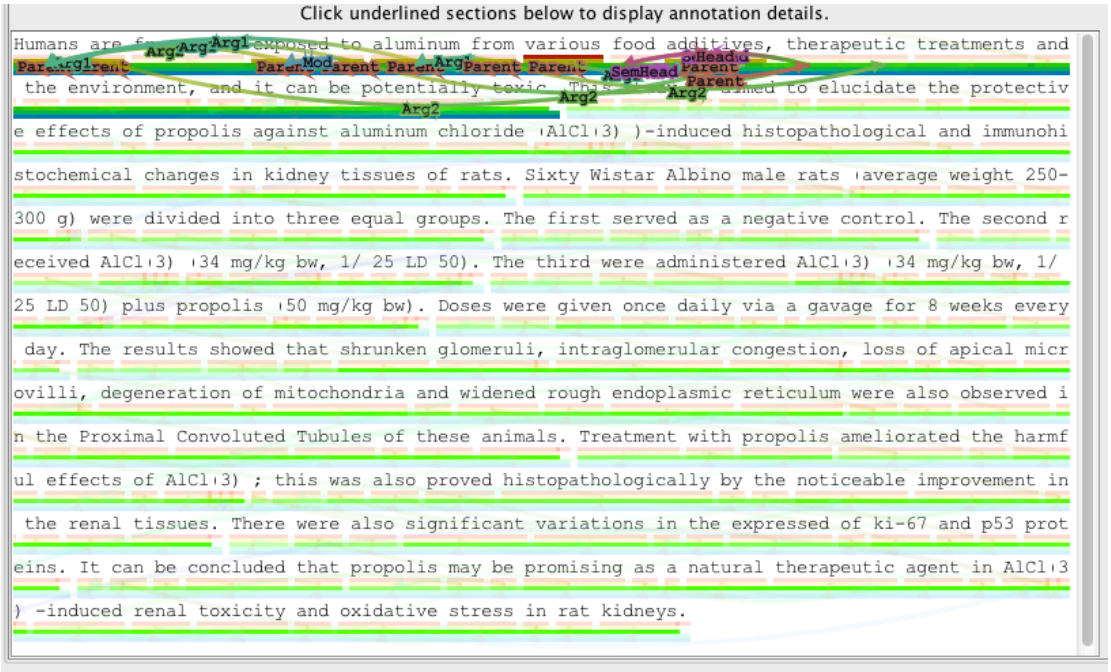


Figure 1: Output of the Enju parser in the U-Compare workbench

Running the tool on the 1 KB text on a single core machine with 8 GB RAM takes around 9738 milliseconds.

3. LICENCES

a) The UIMA wrapper code is licensed using the NaCTeM Software Licence Agreement (standard non-commercial use) – see "Enju-U-Compare-licence.pdf" in the "licences" directory. Please contact us using the details below if you require a commercial licence.

b) The underlying Enju parser web service called by the UIMA code is licensed using the NaCTeM Web Service Licence Agreement (standard non-commercial use)– see "Enju-web-service-licence.pdf" in the "licences" directory. Please contact us using the details below if you require a commercial licence.

c) The UIMA framework is licenced using the Apache licence. Please see "Apache.txt" in the "licences" directory.

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.*