Documentation for

JULIE Lab UIMA Wrapper for OpenNLP Parser

Ekaterina Buyko
Jena University Language & Information Engineering (JULIE) Lab
Fürstengraben 30
D-07743 Jena, Germany
{buyko}@coling-uni-jena.de

1 Objective

JULIE LAB UIMA WRAPPER FOR OPENNLP PARSER is part of the Julie Lab NLP tool suite[1] which contains several NLP components (all UIMA compliant) from sentence splitting to named entity recognition and normalization as well as a comprehensive UIMA type system. The OPENNLP PARSER[2] provides consituency-based parsing. For more detailed information on the functioning of the OPENNLP PARSER check `http://www.opennlp.org`.

UIMA WRAPPER FOR OPENNLP PARSER is currently available in version 2.1.

2 Requirements and Dependencies

JULIE LAB UIMA WRAPPER FOR OPENNLP PARSER is written in Java 1.6 using Apache UIMA version 2.1.0-incubation[3]. It was not tested with other UIMA versions.

The input and output of an AE takes place by annotation objects. The classes corresponding to these objects are part of a *Julie Lab UIMA type systems*[4].

The wrapper comes as a UIMA pear file. Run the Pear-Installer (e.g., `./runPearInstaller.sh` for Linux) from your UIMA-bin directory. After installation, you will find a subfolder `desc`

[1] `http://www.julielab.de/`
[2] `http://www.opennlp.org`
[3] `http://incubator.apache.org/uima/`
[4] The *Julie Lab UIMA type systems* can be obtained from `http://www.julielab.de/`

in your installation folder. This directory contains a descriptor `ParseAnnotator.xml`. You may now e.g. run UIMA's Collection Proeccessing Engine Configurator (`cpeGUI.sh`) and add the wrapper as a component into your NLP pipeline.

This pear package also contains models for parsing. The models are trained on two bio-medical corpora respectively: GENIA ([OTK02]) and PennBIoIE[5]. An accuracy of 86.1% is yielded on the GENIA using 10-fold cross-validation [BWPH06]. You will find the models in the directory `resources`.

3 Using the AE – Descriptor Configuration

In UIMA, each component is configured by a descriptor in XML. In the following we describe how the descriptor required by this AE can be created with *Component Descriptor Editor*, an Eclipse plugin which is part of the UIMA SDK.

A descriptor contains information on different aspects. The following subsection refers to each sub aspect of the descriptor which is, in the Component Descriptor Editor, a separate *tabbed page*. For an indepth description of the respective configuration aspects or tabs, please refer to the *UIMA SDK User's Guide*[6], especially chapter 12 on "Component Descriptor Editor User's Guide".

To define your descriptor go through each tabbed pages mentioned here, make your respective entries (especially in page *Parameter Settings* you will be able to configure OPENNLP PARSER to your needs) and save the descriptor as `DescriptorName.xml`.

Overview This tab provides general informtion about the component. For the OpenNLP Parser you need to provide the information as specified in Table 1.

Aggregate Not needed here, as this AE is a primitive.

Parameters See Table 2 for a specification of the configuration parameters of this AE. Do not check "Use Parameter Groups" in this tab.

Parameter Settings The specific parameter settings are filled in here. For each of the parameters defined in 3, add the respective values here (has to be done at least for each parameter that is defined as mandatory). See Table 3 for the respective parameter settings of this AE.

[5] `http://bioie.ldc.upenn.edu/`
[6] `http://incubator.apache.org/uima/`

Subsection	Key	Value
Implementation Details	Implementation Language	JAVA
	Engine Type	Primitive
Runtime Information	updates the CAS	yes
	multiple deployment allowed	yes
	outputs new CASes	no
	Name of the Java class file	`de.julielab.jules.ae.opennlp.ParseAnnotator`
Overall Identification Information	Name	JULES-OPENNLP-PARSER-AE
	Version	2.1
	Vendor	julielab
	Description	see above

Table 1: Overview/General Settings for AE.

Type System On this page, go to *Imported Type* and add the *julie-morpho-syntax-types.xml* type system. (Use "Import by Location").

Capabilities See Table 4

Index Nothing needs to be done here.

Resources Nothing needs to be done here.

4 Copyright and License

Parameter Name	Parameter Type	Mandatory	Multivalued	Description
modelDir	String	yes	no	Path to the directory with OPENNLP PARSER models
tagset	String	sey	no	Cas type to annotate (see *JULIE UIMA type system)*
useTagdict	Boolean	yes	no	`true` if a tag dictionary should be used
caseSensitive	Boolean	no	no	`true` if a tag dictionary is case senstive
mappings	String (multi-valued)	yes	yes	Mappings between constituent categories provided by the OPENNLP PARSER and *Constituent* feature *cat* (see *jules-morpho-syntpes.xml*)
beamSize	Integer	no	no	Beam size
advancePercentage	String	no	no	Amount of probability mass required of advanced outcomes
fun	Boolean	no	no	`true` if parsing with functional tags (e.g. subj, obj)

Table 2: Parameters of this AE.

Parameter Name	Parameter Syntax	Example
modelDir	String	resources/modelsGenia
tagset	CAS sub-type of *Constituent*	de.julielab.jules.types.GENIAConstituent
useTagdict	Boolean	`true`
mappings	OpenNLP Name;CAS Name	S;S

Table 3: Parameter settings of this AE.

Type	Input	Output
de.julielab.jules.types.Sentence	✓	
de.julielab.jules.types.Token	✓	
de.julielab.jules.types.Constituent		✓

Table 4: Capabilities of this AE.

References

[BWPH06] Ekaterina Buyko, Joachim Wermter, Michael Poprat, and Udo Hahn. Automatically adapting an NLP core engine to the biology domain. In Hagit Shatkay, Lynette Hirschman, Alfonso Valencia, and Christian Blaschke, editors, *Proceedings of the Joint BioLINK-Bio-Ontologies Meeting. A Joint Meeting of the ISMB Special Interest Group on Bio-Ontologies and the BioLINK Special Interest Group on Text Data Mining in Association with ISMB*, pages 65–68. Fortaleza, Brazil, August 5, 2006, 2006.

[OTK02] Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim. The GENIA corpus: An annotated research abstract corpus in molecular biology domain. In M. Marcus, editor, *HLT 2002 – Human Language Technology Conference. Proceedings of the 2nd International Conference on Human Language Technology Research*, pages 82–86. San Diego, Cal., USA, March 24-27, 2002. San Francisco, CA: Morgan Kaufmann, 2002.