Documentation for

JULIE Lab UIMA Wrapper for OpenNLP Tokenizer

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

1 Objective

The OpenNLP Tokenizer[1] identifies tokens in sentences. Julie Lab UIMA Wrapper for OpenNLP Tokenizer is part of the Julie Lab NLP tool suite[2] 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.

2 Requirements and Dependencies

Julie Lab UIMA Wrapper for Tokenizer is written in Java 1.5 using Apache UIMA version 2.2.0-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` in you installation folder. This directory contains a descriptor `TokenAnnotator.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.

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

This pear package also contains two models for tokenization. The models was trained on two bio-medical corpora GENIA ([OTK02]). and PennBIoIE[5] respectively. An accuracy of 99.6% is yielded on the GENIA using 10-fold cross-validation. 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 SKD 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 Tokenizer to your needs) and save the descriptor as `TokenAnnotator.xml`.

Overview This tab provides general informtion about the component. For the OpenNLP Sentence Splitter 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.

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

[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.TokenAnnotator`
Overall Identification Information	Name	JULIELAB UIMA Wrapper for OpenNLP Tokenizer
	Version	2.1
	Vendor	julielab
	Description	see above

Table 1: Overview/General Settings for AE.

Parameter Name	Parameter Type	Mandatory	Multivalued	Description
modelFile	String	yes	no	The path to the OpenNLP Tokenizer model.

Table 2: Parameters of this AE.

Parameter Name	Parameter Syntax	Example
modelFile	model.bin.gz	resources/TokenizerGenia.bin.gz

Table 3: Parameter settings of this AE.

Capabilities Nothing needs to be done here.

Index Nothing needs to be done here.

Resources Nothing needs to be done here.

4 Copyright and License

References

[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.