

Documentation for

# UIMA Wrapper for JULIE Lab Part of Speech Tagger

Version 1.0

Johannes Hellrich  
Jena University Language & Information Engineering (JULIE) Lab  
Fürstengraben 30  
D-07743 Jena, Germany  
`johannes.hellrich@uni-jena.de`

## 1 Objective

The UIMA Wrapper for JULIE Lab Named Entity Tagger (UIMA-JPOS) is an UIMA wrapper for the JULIE Lab Part of Speech Tagger (JPOS). It is part of the JULIE Lab NLP tool suite<sup>1</sup> 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.

For annotating tokens with their part of speech, this analysis engine employs the JULIE Lab Part of Speech Tagger (JPOS). JPOS uses a machine learning (ML) approach, generating (ML-)features in order to select POS tags for a given text of written natural language. JPOS offers the possibility to configure the feature generation. As JPOS needs a UIMA pipeline providing sentence and token annotations in its CAS. It then modifies the token annotations by adding POS tags.

## 2 About this documentation

This is a documentation on using the UIMA-compliant version of JPOS. UIMA-JPOS is a wrapper to JPOS, which actually does all the named entity recognition. To get more information on JPOS itself, please refer to its documentation.

---

<sup>1</sup><http://www.julielab.de/>

## 3 Changelog

1.0 Initial release.

## 4 Requirements and Dependencies

UIMA-JPOS is completely written in Java using Apache UIMA <sup>2</sup>. It requires Java 1.7 (or above).

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 System*.<sup>3</sup> When referring to UIMA annotation types we mean types from the JULIE Lab UIMA type system.

## 5 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 (or modified) with the *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*, especially the chapter 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 UIMA-JPOS to your needs) and save the descriptor as `SomeName.xml`.

**Overview** This tab provides general information about the component. For the UIMA-JPOS 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.

---

<sup>2</sup><https://uima.apache.org>

<sup>3</sup>The *JULIE UIMA type system* can be obtained separately from <http://www.julielab.de/>. However, the necessary parts of the type system are already contained in this package.

Subsection	Key	Value
Implementation Details	Implementation Language	Java
	Engine Type	primitive
Runtime Information	updates the CAS	check
	multiple deployment allowed	check
	outputs new CASes	don't check
	Name of the Java class file	<code>de.julielab.jules.ae.postagger.POSAnnotator</code>
Overall Identification Information	Name	UIMA-JPOS
	Version	1.0
	Vendor	JULIE Lab
	Description	you may keep this empty

Table 1: Overview/General Settings for AE.

Parameter Name	Parameter Type	Mandatory	Multivalued	Description
ModelFilename	String	yes	no	specifies which model JPOS should use
tagset	String	yes	no	specifies which POS tag set to use

Table 2: Parameters of this AE.

**Parameter Settings** The specific parameter settings are filled in here. For each of the parameters defined in 5, 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.

Parameter Name	Parameter Syntax	Example
ModelFilename	Give either the complete path to the model file	<code>/path/to/model</code> or only its <b>name if it resides in the classpath</b>
tagset	full name of the POS tag set used	<code>some.pos.TagSet</code>

Table 3: Parameter settings of this AE.

**Type System** On this page, go to *Imported Type* and import "julie-all-types" by name.

**Capabilities** UIMA-JPOS needs as input annotations from type `de.julielab.jules.types.Sentence` and `de.julielab.jules.types.Token`. It modifies the annotations from type `de.julielab.jules.types.Token`.

Type	Input	Output
<code>de.julielab.jules.types.Sentence</code>	✓	
<code>de.julielab.jules.types.Token</code>	✓	✓

Table 4: Capabilities of this AE.

**Index** Nothing needs to be done here.

**Resources** Nothing needs to be done here.

## 6 Modifying the Descriptors

This PEAR package contain one descriptor for UIMA-JPOS configured for tagging German biomedical texts. We also provide a model for German newspaper text. You can train other models using the JPOS command-line tool; usig JPOS for English is not advised.

## 7 Copyright and License

This software is Copyright (C) 2015 Jena University Language & Information Engineering Lab (Friedrich-Schiller University Jena, Germany), and is licensed under the terms of the Common Public License, Version 1.0 or (at your option) any subsequent version.

The license is approved by the Open Source Initiative, and is available from their website at <http://www.opensource.org>.