Documentation for

JULIE Lab Sentence Boundary Detector

Version 2.2

Katrin Tomanek
Jena University Language & Information Engineering (JULIE) Lab
Fürstengraben 30
D-07743 Jena, Germany
katrin.tomanek@uni-jena.de

1 Objective

The JULIE Lab Sentence Boundary Detector (UIMA-JSBD) is a sentence boundary detector for UIMA. It is part of the JULIE Lab NLP tool suite[1] which contains several UIMA-compliant NLP components from sentence splitting to named entity recognition and normalization as well as a comprehensive UIMA type system.

UIMA-JSBD is is an UIMA wrapper for JSBD, the respective command-line version. For more detailed information on the functioning of JSBD check the JSBD documentation or refer to [TWH07].

2 Installation

UIMA-JSBD 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 `SentenceAnnotator.xml` for UIMA-JSBD. You may now e.g. run UIMA's Collection Proeccessing Engine Configurator (`cpeGUI.sh`) and add UIMA-JSBD as a component into your NLP pipeline.

This pear package also contains a model for sentence splitting. The model was trained on a special bio-medical corpus which consists of data from both the GENIA [OTK02] and the PennBioIE[2] corpus and additional material which we took from MedLine abstracts.

[1] `http://www.julielab.de/`
[2] `http://bioie.ldc.upenn.edu/`

Currently, it comprises about 62000 sentences. An accuracy of 99.8% is yielded on this data using 10-fold cross-validation. You will find the model trained on this data in the directory `resources`.

3 Requirements and Dependencies

UIMA-JSBD is written in Java (version 1.5 or above required) using Apache UIMA version 2.2.1-incubation[3].

The input and output of an AE takes place by annotation objects. The classes corresponding to these objects are part of the *JULIE Lab UIMA Type System* in its current version (2.1).[4]

This version of UIMA-JSBD is based on JSBD-1.6 which employs the machine learning toolkit MALLET [McC02].

4 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 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 SKD User's Guide*[5], especially the chapter on "Component Descriptor Editor User's Guide".

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

Otherwise, you can of course employ the descriptor that is contained in the pear package you downloaded (in your installation directory, see `desc/SentenceAnnotator.xml`).

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

[3] `http://incubator.apache.org/uima/`

[4] The *JULIE Lab UIMA type system* can be separately obtained from `http://www.julielab.de/`, however, this package already includes the necessary parts of the type system.

[5] `http://incubator.apache.org/uima/`

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	`de.julielab.jules.ae.SentenceAnnotator`
Overall Identification Information	Name	Sentence Annotator
	Version	2.2
	Vendor	JULIE Lab
	Description	not needed

Table 1: Overview/General Settings for AE.

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 4, 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 following layers of the *JULIE UIMA Type System* (Use "Import by Location"): `julie-basic-types.xml` and `julie-morpho-syntax-types.xml`. If you use the *ProcessingScope* parameter make sure that the respective type/type system is also included.

Capabilities The sentence splitter only returns annotations from type `de.julielab.jules.types.Sentence`. See Table 4.

Index Nothing needs to be done here.

Parameter Name	Parameter Type	Mandatory	Multivalued	Description
ModelFilename	String	yes	no	filename of trained model for JSBD
Postprocessing	Boolean	no	no	Indicates whether post-processing should be run. Default: no post-processing
ProcessingScope	String	no	no	The UIMA annotation type over which to iterate for doing the sentence segmentation. If nothing is given, the document text from the CAS is taken as scope! This is recommended as default!

Table 2: Parameters of this AE.

Parameter Name	Parameter Syntax	Example
ModelFilename	full path	`resources/JULIE_life-science-1.6.mod.gz`
Postprocessing	true/false	true
ProcessingScope	full class name to annotation type	`de.julielab.jules.paragraph` (assuming you downloaded the document structure part of the JULIE Lab Type System). If you don't know what to do here, leave it blank!

Table 3: Parameter settings of this AE.

Type	Input	Output
de.julielab.jules.types.Sentence		√

Table 4: Capabilities of this AE.

Resources Nothing needs to be done here.

5 Copyright and License

References

[McC02] Andrew McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[OTK02] Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim. The GENIA corpus: An annotated research abstract corpu s 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: Mo rgan Kaufmann, 2002.

[TWH07] Katrin Tomanek, Joachim Wermter, and Udo Hahn. A reappraisal of sentence and token splitting for life science documents. In *MEDINFO 2007 – Proceedings of the 12th World Congress on Me dical Informatics.*, 2007.