Documentation for

JULIE Lab Acronym Annotator

Version 2.1

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

1 Objective

JULIE Lab Acronym Annotator (JACRO) is an UIMA Analysis Engine that annotates acronyms with their full-forms when locally introduced in the current document. 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.

The functionality of the engine is based on the simple algorithm for abbreviation recognition by Schwartz and Hearst[2]. We have reimplemented the algorithm and extended it with respect to some pattern definitions and normalizations.

During the processing of documents, this UIMA annotator takes sentence annotions from the CAS and creates an `Abbreviation` annotation object for each identified acronym. The `Abbreviation` annotation stores the corresponding full-form, whether the acronym was introduced at the respective position, and a reference to the full-form in the text.

2 Requirements and Dependencies

The annotator is completely written in Java (at least Java 1.5 required) using Apache UIMA version 2.2.1-incubation[3].

[1] `http://www.julielab.de/`
[2] `http://biotext.berkeley.edu/papers/psb03.pdf`
[3] `http://incubator.apache.org/uima/`

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]

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 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 descriptor go through each tabbed page mentioned here, make your respective entries and save the descriptor as e.g. `AcronymAnnotatorDescriptor.xml`.

As this package already contains a pre-configured descriptor (see `desc/AcronymAnnotator.xml`) there is no need to build such a descriptor from scratch. However, you might modify the parameter settings according to your needs.

Overview This tab provides general informtion about the component. For the Acronym Annotator 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 Table 2, 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.

[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	yes
	multiple deployment allowed	yes
	outputs new CASes	yes
	Name of the Java class file	`de.julielab.jules.ae.acronymtagger.AcronymAnnotator`
Overall Identification Information	Name	Acronym Annotator
	Version	2.1
	Vendor	julielab
	Description	you may keep this empty

Table 1: Overview/General Settings for AE.

Parameter Name	Parameter Type	Mandt.	Multi-valued	Description
ConsistencyAnno	Bool	yes	no	specifies whether only the first or all occurences of the acronym are annotated in the document
MaxLength	Integer	yes	no	Define how far (how many words, ignoring stopwords) the AE is supposed too look for the beginning of the full-form.

Table 2: Parameters of this AE.

Parameter Name	Parameter Syntax	Example
ConsistencyAnno	set to true iff you want to annotate all occurences of the found acronyms	true
MaxLength	just an integer	5

Table 3: 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`.

Capabilities JACRO takes as input annotations from type `de.julielab.jules.types.Sentence` and returns annotations from type `de.julielab.jules.types.Abbreviation`. See Table 4.

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

Table 4: Capabilities of this AE.

Index Nothing needs to be done here.

Resources Nothing needs to be done here.

4 Copyright and License