

Documentation for

# JULIE Lab Lingpipe Gazetteer Annotator

Version 2.1

Katrín Tomanek                      Joachim Wermter

Jena University Language & Information Engineering (JULIE) Lab

Fürstengraben 30

D-07743 Jena, Germany

`joachim.wermter@uni-jena.de`

## 1 Objective

JULIE Lab Lingpipe Gazetteer Annotator is an UIMA Analysis Engine that annotates document text entities with their dictionary matches as specified in the given dictionary file. It is part of the JULIE Lab NLP tool suite<sup>1</sup> which contains several UIMA-compliant NLP components from sentence splitting to named entity recognition and normalization as well as a comprehensive UIMA type system.

This engine uses Lingpipe's Exact and Approximate Dictionary-Based Chunking which is based on implementation of the Aho-Corasick algorithm ([http://en.wikipedia.org/wiki/Aho-Corasick\\_algorithm](http://en.wikipedia.org/wiki/Aho-Corasick_algorithm)) which finds all matches against a dictionary in linear time independent of the number of matches or size of the dictionary.

During the processing of documents, this UIMA annotator takes the whole text annotation from the CAS and creates an **EntityMention** annotation object for each identified dictionary entry. A different annotation type object may be specified in the descriptor. Besides the text offsets, the **EntityMention** annotation stores the **specificType** of a matched dictionary entry, which may be a normalized form (e.g. a database identifier) or some intermediate entity type (e.g. "Gene").

---

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

## 2 Requirements and Dependencies

The annotator is completely written in Java (at least Java 1.6 required) using Apache UIMA version 2.2.1-incubation<sup>2</sup>.

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).<sup>3</sup>

## 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 SDK User's Guide*<sup>4</sup>, 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. `GazetteerAnnotatorDescriptor.xml`.

**As this package already contains a pre-configured descriptor (see `desc/GazetteerAnnotatorTest.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 information 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.

---

<sup>2</sup><http://incubator.apache.org/uima/>

<sup>3</sup>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.

<sup>4</sup><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.lingpipegazetteer.GazetteerAnnotator
Overall Identification Information	Name	Gazetteer 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
UseApproximateMatching	Bool	yes	no	specifies whether matching should be done approximately
IgnoreCase	Bool	yes	no	specifies whether case should be ignored while matching.
DictionaryFile	String	yes	no	Path to dictionary file..
OutputType	String	yes	no	The output type where dict matches are written to (default is de.julielab.jules.EntityMention).

Table 2: Parameters of this AE.

**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.

**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`, `julie-morpho-syntax-types.xml`, and `julie-semantics-mention-types.xml`.

Parameter Name	Parameter Syntax	Example
UseApproximateMatching	set to true iff you want to match approximately. Notice that this slows down running time significantly	true
IgnoreCase	set to true iff you want to match case-insensitively	true
DictionaryFile	Dictionary file needs to be in formatted as such: name1 TAB entityType1 name2 TAB entityType2 (for an example, see: resources/dictionary.tst)	resources/dictionary.tst

Table 3: Parameter settings of this AE.

**Capabilities** The Gazetteer Annotator takes as input annotations the whole document text from the CAS (hence needs not to be specified in the Capabilities). By default, it returns annotations from type `de.julielab.jules.types.EntityMention`, which may be overridden in the descriptor. See Tables 3 and 4.

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

Table 4: Capabilities of this AE.

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

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

## 4 Copyright and License

The GazetteerAnnotator is Copyright (C) 2008 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. This license is approved by the Open Source Initiative, and is available from their website at <http://www.opensource.org>.

The Lingpipe Core Library is available under various licenses, as specified under <http://alias-i.com/lingpipe/web/download.html>. The one included here is the Alias-i

ROYALTY FREE LICENSE VERSION 1 (<http://alias-i.com/lingpipe/licenses/lingpipe-license-1.txt>). If you wish to receive a license from Alias-i under different terms than those contained in this License, please contact Alias-i.

**ATTENTION: Please make sure to notice that this UIMA Annotator pear package (jules-lingpipe-gazetteer-ae-1.2.pear) and its core Lingpipe dependency (lingpipe-3.3.0.jar) are not distributed under the same license!**