

Documentation for

JULIE Lab MUC7 Collection Reader

Michael Poprat

Jena University Language & Information Engineering (JULIE) Lab

Fürstengraben 30

D-07743 Jena, Germany

`michael.poprat@uni-jena.de`

1 Objective

The JULIE LAB MUC7 COLLECTION READER is an UIMA Collection Reader (CR). It reads MUC7 data and converts it to types defined the UIMA type systems that we provide as well. In particular, all paragraphs are read and their offset is calculated. The same holds for the MUC7 named entities as well as for the annotated coreferences. However, the event annotation (that are the templates in BNF format) are not processed, yet.

The JULIE LAB MUC7 COLLECTION READER is part of the JULIE NLP tool suite¹ 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.

The JULIE LAB MUC7 COLLECTION READER is currently available in version 1.1. For more detailed information about the MUC7 data, please read [MUC98].

2 Requirements and Dependencies

The JULIE LAB MUC7 COLLECTION READER is written in Java 1.6 using Apache UIMA version 2.2.1-incubation². It was not tested with other UIMA versions.

¹<http://www.julielab.de/>

²<http://incubator.apache.org/uima/>

The input of the JULIE LAB MUC7 COLLECTION READER are the MUC7 data files that can be purchased at the Linguistic Data Consortium (LDC)³.

However, the MUC7 files are only provided in SGML format and must be made XML compatible in advance before they can be read by the CR. In particular the paragraphs must be closed (<p>) and some special characters must be removed or escaped.

The output of the CR takes place by annotation objects. The classes corresponding to these objects are part of the *JULIE UIMA Type System*⁴.

The CR 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 `MUC7ReaderDescriptor.xml`. For instance you can implement this reader in a pipeline in order to further process the text with a tokenizer and a sentence splitter. Or you can use the annotated data to train classifiers for named entities or resolvers for coreference resolution.

3 Using the CR – Descriptor Configuration

In UIMA, each component is configured by a descriptor in XML. In the following we describe how the descriptor required by this CR 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*⁵, 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 the JULIE LAB MUC7 COLLECTION READER to your needs) and save the descriptor as `MUC7ReaderDescriptor.xml`.

Overview This tab provides general information about the CR. For the JULIE LAB MUC7 COLLECTION READER you need to provide the information as specified in Table 1.

Aggregate Not needed here, as this CR is a primitive.

³<http://www ldc upenn edu/>

⁴The *JULIE UIMA type systems* can be obtained from <http://www julielab de/>

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	<code>de.julielab.jules.reader.MUC7Reader</code>
Overall Identification Information	Name	JULIE LAB MUC7 COLLECTION READER
	Version	1.1
	Vendor	julielab
	Description	see above

Table 1: Overview/General Settings for CR.

Parameters See Table 2 for a specification of the configuration parameters of this CR. Do not check “Use Parameter Groups” in this tab.

Parameter Name	Parameter Type	Mandatory	Multivalued	Description
InputDirectory	String	yes	no	path to the MUC7 files

Table 2: Parameters of this CR.

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 CR.

Parameter Name	Parameter Syntax	Example
InputDirectory	<code>/path/to/MUC7/files</code>	<code>resources/MUC7data</code>

Table 3: Parameter settings of this CR.

Type System On this page, go to *Imported Type* and add the following type systems:

- *julie-basic-types.xml*

- *julie-semantics-mention-types.xml*
- *julie-muc7-types.xml*
- *julie-document-meta-types.xml*
- *julie-document-structure-types.xml*
- *julie-morpho-syntax-types.xml*

Use “Import by Location”.

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

This software 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 2.1.

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

References

- [MUC98] MUC-7. *Proceedings of the 7th Message Understanding Conference, NYU*. 1998.