

Documentation for
JULIE Lab's Wikipedia Reader
version 0.2.4

Elena Beisswanger
Jena University Language & Information Engineering (JULIE) Lab
Fürstengraben 30
07743 Jena, Germany
elena.beisswanger@uni-jena.de

Contents

1	Introduction	2
1.1	Using the WIKIPEDIA Reader– Step by Step Guide	2
1.2	Text Cleansing	2
1.3	Composing and Annotating Text	3
2	Requirements and Dependencies	4
2.1	The WIKIPEDIA Database	4
2.1.1	How to Setup the Database	4
2.1.2	Database Contents	5
2.1.3	Initializing / Resetting the WIKIPEDIA Database	5
2.2	Type System Dependencies	6
2.3	JWPL Dependencies	6
3	The Wikipedia Reader Component Descriptor	6
3.1	Overview	6
3.2	Parameters	7
3.3	Parameter Settings	8
3.4	Type System	8
3.5	Important Notes on Configuration Parameters	9
4	Embedding in a Scheduling System	9
4.1	Status Callback Listener	10
5	Availability, Copyright and License	10

1 Introduction

The WIKIPEDIA Reader is a UIMA Collection Reader for WIKIPEDIA. It is part of the JULIE NLP tool suite which contains a set of UIMA compliant NLP components and a comprehensive UIMA Type System [2]. The JULIE NLP tools are available from our website.

When the WIKIPEDIA Reader is included as collection reader in an UIMA pipeline, this is how it operates: It reads in those WIKIPEDIA pages from a database that are marked for processing (see 2.1.1 to learn how to set up the WIKIPEDIA database, and 2.1.3 to learn how to mark pages for processing). For each page it preprocesses the page text (see 1.2), parses the MEDIAWIKI mark-up, composes a markup-free version of the text and creates annotations marking text fractions as paragraph, list, caption, etc. (see 1.3), and adds the reassembled document text together with the annotations to a new CAS. The CAS becomes the UIMA internal representation of the WIKIPEDIA page. The WIKIPEDIA Reader is configurable via an XML-based UIMA component descriptor (see 3).

1.1 Using the Wikipedia Reader– Step by Step Guide

1. Download the WIKIPEDIA Reader PEAR package from our website and install it as described in the tutorial “Getting Started: Working With PEARS”.
2. Setup a WIKIPEDIA database as described in 2.1.1.
3. Create an UIMA component descriptor for the WIKIPEDIA Reader, e.g. by adapting the descriptor file contained in the “desc” folder of the WIKIPEDIA Reader PEAR package. The descriptor essentially defines parameter settings and type system dependencies (see 3).
4. Setup your UIMA pipeline and provide it with the WIKIPEDIA Reader descriptor.
5. Create a listener component complementing the WIKIPEDIA Reader as described in 4.1.
6. Before you rerun your pipeline, reset the WIKIPEDIA database as described in 2.1.3.

1.2 Text Cleansing

The WIKIPEDIA Reader processes the WIKIPEDIA page text prior to and after parsing the MEDIAWIKI mark-up to render a properly processable document text to subsequent text analytics (incorporating other UIMA components such as sentence splitters or syntax parsers).

Prior to the MEDIAWIKI mark-up parsing step (in which all tags embedded in angled brackets will be removed), text enclosed in “<ref>”¹ and “<timeline>”² tags is dropped by the reader since it presents special, from our perspective inappropriate, contents.

After the parsing step, yet before the text of particular content items of a WIKIPEDIA page becomes part of the document text, it is cleansed and adapted by the reader. Empty brackets and quotes, invalid XML characters, as well as leading and trailing whitespaces are removed, and duplicate whitespaces and non-breaking spaces are replaced by single whitespaces. If the text ends with one of “{,;!?}”, the last character is replaced by a full stop. If it does not end with one of “{,;!?}” (as, e.g., many list or table elements do), a full stop is appended. This is to prevent any sentence splitter component to create very long sentences, e.g. by interpreting text derived from a flattened list or table as one sentence. Concatenated text chunks are separated by single whitespaces. Depending on the last non-whitespace character in the document text, the next chunk starts with an upper case or a lower case letter (upper case only in case that one of “{,;!?}” is encountered).

1.3 Composing and Annotating Text

After parsing the MEDIAWIKI mark-up the parser incorporated in the WIKIPEDIA Reader outputs Java objects corresponding to the different content items of the WIKIPEDIA page (sections, paragraphs, lists, tables, and image captions). The reader handles these content items in consecutive order. From each item the text is extracted, cleansed (see 1.2) and added to the CAS document text. If the parameter **SkipTableContents** is switched on, table bodies are omitted and only table captions are considered. To preserve the information from which content item a particular text fragment has been derived from, an appropriate UIMA annotation (defined in the *JULIE UIMA Type System*, see 2.2) is attached to it. For example, to the text derived from a paragraph an annotation of type **de.julielab.jules.types.Paragraph** is attached. In addition, for all WIKIPEDIA internal links occurring in any of the text segments **de.julielab.jules.types.wikipedia.Link** annotations are created holding the link target as value of the target feature. This is the full list of annotation types used by the WIKIPEDIA Reader:

- **de.julielab.jules.types.Header** (with features for article ID and title)
- **de.julielab.jules.types.wikipedia.Descriptor** (with features to carry categories, incoming links, outgoing links and redirects of a WIKIPEDIA page)
- **de.julielab.jules.types.wikipedia.ArticleText**

¹These tags mark references and belong to the MEDIAWIKI “Cite” extension.

²These tags mark wikitext from which embedded images are produced. They belong to the “Easy-Timeline” extension.

- `de.julielab.jules.types.wikipedia.Title` (with features for the full WIKIPEDIA page name, i.e., including disambiguation part, if available, and the short version of it)
- `de.julielab.jules.types.Section` (with a feature to carry a list of text objects, such as tables and images)
- `de.julielab.jules.types.Paragraph`
- `de.julielab.jules.types.TextObject` (with features for specifying the object type, e.g. “table” or “image”, and a caption)
- `de.julielab.jules.types.Caption` (used for annotating image and table captions)
- `de.julielab.jules.types.List` (with a feature to carry a list of list item annotations)
- `de.julielab.jules.types.ListItem`
- `de.julielab.jules.types.wikipedia.Link` (with a link target feature)

2 Requirements and Dependencies

The WIKIPEDIA Reader is written in Java 1.5 using Apache UIMA version 2.2.2-incubating [1]. It has not been tested with other UIMA versions.

2.1 The Wikipedia Database

The WIKIPEDIA Reader, by design, does not access WIKIPEDIA online, but reads in WIKIPEDIA information from a database.

2.1.1 How to Setup the Database

To set up the database, proceed as follows:

1. Create an empty MySQL database. (Note that we only used the WIKIPEDIA Reader in combination with a MySQL database and the default storage engine MyISAM. In case another database is used, especially in combination with a storage engine supporting transactions, a code adaptations of the reader might be required.)
2. Download the JWPLDataMachine tool from UKP Lab’s Java WIKIPEDIA Library (JWPL) website.
3. Follow the instructions given in the JWPL tutorial, i.e., download the appropriate WIKIPEDIA XML dumps (containing the last revision of pages only) and use the JWPLDataMachine to generate a WIKIPEDIA SQL dump out of it.

4. Import the SQL dump in the database. This will generate the appropriate table structure and populate the tables.
5. Extend the `Page` table by adding the following new fields: `include`, `is_processed`, `is_in_process`, `has_errors`, `log`, `host_name` and `pid`. The following SQL command generates the new fields:

```
ALTER TABLE Page ADD COLUMN is_processed BIT NOT NULL DEFAULT FALSE, ADD
COLUMN is_in_process BIT NOT NULL DEFAULT FALSE, ADD COLUMN include BIT
NOT NULL DEFAULT TRUE, ADD COLUMN has_errors BIT NOT NULL DEFAULT FALSE,
ADD COLUMN log TEXT, ADD COLUMN host_name VARCHAR(100), ADD COLUMN pid
VARCHAR(10);
```

2.1.2 Database Contents

The JWPLDataMachine creates a number of tables, not all of them are required by the WIKIPEDIA Reader.

The main table of the WIKIPEDIA database is named `Page`. It holds ID, name and the complete page text (written in MEDIAWIKI mark-up) of all WIKIPEDIA pages in the article namespace (i.e., proper article pages, disambiguation pages, redirect pages and lists), excluding the redirect pages. (Note that redirect links, instead, are contained in a separate table named `page_redirects`.) In addition, the `Page` table contains the field `isDisambiguation`. The JWPLDataMachine sets the value of this field to true for all disambiguation pages that it has recognized as such. The `include` field was added to allow for marking WIKIPEDIA pages for inclusion / exclusion for analysis. The fields `is_processed` and `is_in_process` are required to track the processing status of each pages when running one or several NLP pipelines, each with its own reader instance, in parallel. The fields `has_errors` and `log` have been included for debugging purposes. Additional tables needed by the reader hold data on WIKIPEDIA categories (IDs and names), link information and page redirects.

During processing, the WIKIPEDIA Reader reads in data from the `Page` table and updates it consistently. When the parameter `AddMetaData` is set to true, the WIKIPEDIA Reader additionally reads from (but never updates) the tables `page_inlinks`, `page_outlinks`, `page_categories`, `redirects` and `Category`. (Note that in a running pipeline besides the WIKIPEDIA Reader there is another component accessing and writing to the `Page` table: the listener component described in 4.1.)

2.1.3 Initializing / Resetting the Wikipedia Database

Before (re)running one or several pipelines incorporating the WIKIPEDIA Reader, make sure that in the WIKIPEDIA database all pages are marked as unprocessed (set `is_processed` and `is_in_process` to false for all pages) and the right pages are marked for inclusion for analysis. To mark all pages as unprocessed execute the following SQL command:

```
UPDATE Page SET is_processed = FALSE, is_in_process = FALSE, has_errors = FALSE,  
host_name = NULL, pid = NULL;
```

There are two ways to restrict the input of the WIKIPEDIA Reader to a subset of WIKIPEDIA. The first one is to mark the corresponding articles directly in the database by setting the value of the **include** field to true. Alternatively, especially to define smaller fractions of WIKIPEDIA, the configuration parameters **PageList** and **CategoryList** can be used (see Section 3.2, and also read 3.5).

2.2 Type System Dependencies

The annotations created by the WIKIPEDIA Reader are based on UIMA Annotation Types defined in the *JULIE UIMA Type System* [3], version 2.6.8 or higher. All Types the reader requires are included in the WIKIPEDIA Reader UIMA PEAR package. However, the Type System can also be obtained separately from our website.

2.3 JWPL Dependencies

For parsing the MEDIAWIKI mark-up the WIKIPEDIA Reader uses the JWPL Parser. To set-up the WIKIPEDIA database, the JWPLDataMachine tool is used. Both tools are part of the Java WIKIPEDIA Library (JWPL) developed at Ubiquitous Knowledge Processing (UKP) Lab (Technische Universität Darmstadt) [4].

3 The Wikipedia Reader Component Descriptor

Via the UIMA component descriptor configuration parameters are defined, values are assigned to parameters, and type system dependencies are specified, amongst others.

UIMA comes with a tool assisting you in editing the descriptor, called the *Component Descriptor Editor* (CDE). Multiple tabs provide different views on the descriptor. In the following subsections we describe how to edit the different tabs to end up with a complete WIKIPEDIA Reader descriptor. (For an in-depth description of the respective configuration aspects or tabs, please refer to “Chapter 1. Component Descriptor Editor User’s Guide” of the “UIMA Tools Guide and Reference” document.) When the descriptor has been completed, save it as, e.g., “WikipediaReaderDescriptor.xml”.

3.1 Overview

The “Overview” tab provides general information about the component. For the WIKIPEDIA Reader you need to provide the information as specified in Table 1.

Subsection	Key	Value
Runtime Information	updates the CAS	true
	Name of the Java class file	de.julielab.jules.reader.WikipediaReader
Overall Identification Information	Name	WikipediaReader
	Version	0.2.4
	Vendor	julielab
	Description	A UIMA collection reader for WIKIPEDIA

Table 1: CDE Overview Tab: General Settings for the WIKIPEDIA Reader

3.2 Parameters

In the “Parameter” tab parameters of the WIKIPEDIA Reader are specified. Available parameters are listed in table 2. Do not check “Use Parameter Groups” in this tab.

Parameter Name	Type	Man- datory	Multi- valued	Description
DataBaseServerURL	String	yes	no	database server URL
DataBase	String	yes	no	name of the database
DataBaseUser	String	yes	no	database user name
DataBasePassword	String	yes	no	database user password
DataBaseDriver	String	no	no	database driver
BatchSize	Integer	no	no	number of documents fetched from the DB at once
OnlyArticles	Boolean	no	no	omit disambiguation pages
PageList	String	no	yes	pages to be considered (read 3.5!)
CategoryList	String	no	yes	categories to be considered (read 3.5!)
ImageIdentifiers	String	no	yes	WIKIPEDIA image identifiers
CategoryIdentifiers	String	no	yes	WIKIPEDIA category identifiers
SkipTableContents	Boolean	no	no	omit table contents, keeping captions
SectionsToSkip	String	no	yes	omit listed sections
Language	String	no	no	language of the WIKIPEDIA used
AddMetaData	Boolean	no	no	create descriptor annotation (read 3.5!)

Table 2: Parameters of the WIKIPEDIA Reader

3.3 Parameter Settings

In the “Parameter Settings” tab values are assigned to the WIKIPEDIA Reader parameters. Note that parameters need to be defined first (e.g. using the tab “Parameters”, see 3.2) to be visible here. Mandatory parameters need to be assigned a value, otherwise during initialization the WIKIPEDIA Reader throws an exception. Optional parameters may be left unassigned. Table 3 shows default values of all WIKIPEDIA Reader configuration parameters, if available.

Parameter Name	Default
DataBaseServerURL	
DataBase	-
DataBaseUser	-
DataBasePassword	-
DataBaseDriver	com.mysql.jdbc.Driver
BatchSize	100
OnlyArticles	true
PageList	-
CategoryList	-
ImageIdentifiers	“Image”
CategoryIdentifiers	“Category”
SkipTableContents	true
SectionsToSkip	“References”, “External links”, “Further reading”, “See also”, “Footnotes”, “Bibliography”, “Literature”
Language	“en”
AddMetaData	false

Table 3: Parameter settings of the WIKIPEDIA Reader.

3.4 Type System

In the “Type System” tab, on the right hand side (below “Imported Type Systems”) add the path to the following type system files, all being part of the *JULIE UIMA Type System*:

- julie-document-structure-types.xml
- julie-document-meta-types.xml
- julie-wikipedia-types.xml

3.5 Important Notes on Configuration Parameters

If the parameters **PageList** and **CategoryList** are used to restrict the analysis to a subset of WIKIPEDIA, the indicated pages will be processed independent from the values in the **Page** table fields **include**, **is_in_process** and **is_processed** (i.e., when using the parameter **PageList** or **CategoryList**, the scheduling system as described in Section 4 is by-passed).

If the parameter **AddMetaData** is set to true, many additional database queries are made to retrieve page meta data from the database. This might decrease the performance of the reader significantly. Thus only set this parameter to true if you really need the meta data for further processing steps in your pipeline.

4 Embedding in a Scheduling System

In order to process a large document collection such as WIKIPEDIA it is desirable to run several pipelines in parallel, each containing its own instance of the WIKIPEDIA Reader. When multiple reader instances simultaneously access the same WIKIPEDIA database, a scheduling system is required to keep track of the processing status of each WIKIPEDIA page.

We established such a scheduler and made the WIKIPEDIA Reader participating in it. It is based on setting the values of the **Page** table fields **is_in_process** and **is_processed** according to the processing status of each WIKIPEDIA page. Besides the WIKIPEDIA Reader, a listener component attached to the pipeline is required as second component of the scheduler (see Section 4.1). In essence, the scheduler runs as follows:

- First the database needs to be initialized / reset by marking all records in the **Page** table as not being processed or in process (see Section 2.1.3).
- When a pipeline is running, the WIKIPEDIA Reader component queries the database for pages that still need to be processed (i.e. pages with **include** true and **is_in_process** and **is_processed** false). If there are pages to be processed, the reader retrieves them, batch-wise, and sets the value of **is_in_process** to true.
- When all analysis components of the pipeline have terminated processing a CAS, the listener attached to the pipeline keeps track of the termination and sets the value of **is_in_process** to false and of **is_processed** to true.
- When no more pages need to be processed (i.e. when for all pages with **include** true the value of **is_processed** is also true, the pipeline eventually stops.

4.1 Status Callback Listener

To provide a pipeline incorporating the WIKIPEDIA Reader with a fully functional scheduling system, besides the reader an appropriate listener component needs to be attached to it. If you use the WIKIPEDIA Reader in a CollectionProcessingEngine (CPE)-based pipeline, simply implement the StatusCallbackListener interface (see the UIMA tutorial on StatusCallbackListeners). In case of a CPE-based pipeline, when all analysis components of the pipeline have completed processing a CAS, the `entityProcessComplete` method of the listener is called. Within this method the listener should update the Page table record corresponding to the CAS by setting the value of `is_in_process` to false and of `is_processed` to true. Furthermore, if an error has occurred during processing the CAS the value of `has_errors` should be set to true and the corresponding log message should be inserted as value of the `log` field.

5 Availability, Copyright and License

The WIKIPEDIA Reader is available as UIMA PEAR package from our website at <http://www.julielab.de>. The WIKIPEDIA Reader itself is Copyright (C) 2010 Jena University Language & Information Engineering Lab (Friedrich-Schiller University Jena, Germany). It 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>.

However, please note that a different licence applies to the JWPL library used by our WIKIPEDIA Reader. Please contact the UKP Lab for further information. (As of March 2010, JWPL is freely available for academia but not open source yet. According to the developers the library is currently in the process of becoming open source.)

When you use the WIKIPEDIA Reader, please cite:

“JULIE Lab’s UIMA Collection Reader for WIKIPEDIA , Elena Beisswanger and Udo Hahn, Proceedings of the New Challenges for NLP Frameworks workshop at LREC 2010, 22 May 2010, La Valleta, Malta, pp. 15-29, 2010”

References

- [1] David Ferrucci and Adam Lally. UIMA: An architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering*, 10(3-4):327–348, 2004.
- [2] Udo Hahn, Ekaterina Buyko, Rico Landefeld, Matthias Mühlhausen, Michael Poprat, Katrin Tomanek, and Joachim Wermter. An overview of JCoRe, the JULIE lab UIMA component repository. In *Proceedings of the LREC’08 Workshop ‘Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP’*, pages 1–7, Marrakech, Morocco, May 2008.

- [3] Udo Hahn, Ekaterina Buyko, Katrin Tomanek, Scott Piao, John McNaught, Yoshimasa Tsuruoka, and Sophia Ananiadou. An annotation type system for a data-driven NLP pipeline. In *The LAW at ACL 2007 – Proceedings of the Linguistic Annotation Workshop*, pages 33–40. Prague, Czech Republic, June 28-29, 2007. Stroudsburg, PA: Association for Computational Linguistics, 2007.
- [4] Torsten Zesch, Christof Müller, and Iryna Gurevych. Extracting lexical semantic knowledge from Wikipedia and Wiktionary. In *LREC'08 – Proceedings of the 6th International Language Resources and Evaluation Coonference*, Marrakech, Morocco, may 2008. European Language Resources Association (ELRA). <http://www.lrec-conf.org/proceedings/lrec2008/>.