Eliza Margaretha *2015*

IDS Wikipedia Converter Documentation

The wikipedia converter developed at the Institut für Deutsche Sprache (IDS)[1] consists of two tools: WikiXMLConverter and WikiI5Converter (Margaretha and Lüngen, 2014). The purpose of the converter is to convert the contents of wikipedia pages from wikipedia dumps written originally in wikitext format, into wikipedia corpora in I5, the IDS text model currently used in DeReKo (Das Deutsche Referenzkorpus)[2] and COSMAS (Corpus Search, Management and Analysis System)[3]. I5 is a customized TEI format based on XCES, enriched with metadata information on different corpus structure levels (Lüngen and Sperberg-McQueen, 2012).

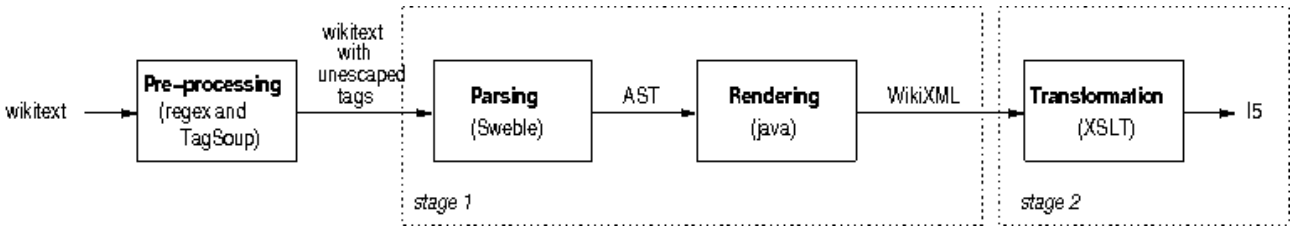

Figure 1 Wikipedia Converter Pipeline

Figure 1 shows the pipeline of the Wikipedia conversion process. The conversion takes wikitext as an input, which is firstly pre-processed using regular expression and the TagSoup parser[4] before entering the conversion stages. In the first stage, the WikiXMLConverter tool converts wikitext into WikiXML by using Sweble Parser[5] and generates a WikiXML file for each wikipage within a wikipedia namespace, for instance articles. In the second stage, WikiI5Converter converts each WikiXML file into I5 using XSLT Stylesheets and assemble them altogether as a single corpus (file).

[1] http://www.ids-mannheim.de/
[2] http://www.ids-mannheim.de/kl/projekte/korpora/
[3] http://www.ids-mannheim.de/cosmas2/
[4] http://home.ccil.org/~cowan/tagsoup/
[5] http://sweble.org/

Table of Contents

1 Availability

The data and tools needed for the Wikipedia corpus conversion can be downloaded from: `http://corpora.ids-mannheim.de/pub/tools/`. The table below describes the files in the folder.

Table 1 Tools and Data for Wikipedia Convertion

Filename	Description
`dewiki-20130728-sample.xml`	A small sample of a German wikipedia dump
`german-inflectives.xml`	A list of German interaction words
`WikiXMLCorpusIndexer.sh`	A shell script for creating a WikiXML corpus index
`2015/WikiXMLConverter-1.0.1-jar-with-dependencies.jar`	The Java library for converting Wikitext to WikiXML version 1.0.1
`2015/WikiXMLConverter-0.0.1-javadoc.jar`	The Java documentation of the WikiXMLConverter code version 1.0.1
`2015/WikiXMLConverter-0.0.1-sources.jar`	Java source code of the WikiXMLConverter version 1.0.1
`2015/WikiI5Converter-0.0.1-jar-with-dependencies.jar`	The Java library for converting WikiXML to I5 converter version 1.0.1
`2015/WikiI5Converter-0.0.1-javadoc.jar`	The Java documentation for WikiI5Converter source code version 1.0.1
`2015/WikiI5Converter-0.0.1-sources.jar`	Java source code of the WikiI5Converter version 1.0.1

The older version of the WikiXMLConverter and WikiI5Converter tools can be found in the `2013/` folder.

2 WikiXMLConverter

To run WikiXMLConverter, a wikipedia dump and a properties file are required. Wikipedia dumps can be downloaded from https://dumps.wikimedia.org/. To convert a full wikipedia (not only a small sample), it is advised to increase and limit the java memory allocation pool by using `-Xmx` parameter. For instance, `-Xmx4g` means set maximum Java heap size to 4 Gigabytes. The following is an example command to run the WikiXMLConverter tool in a terminal.

```
java -jar -Xmx4g [jar-file-path] -prop [properties-file-path] > [log-file-path] 2>&1
```

The main class
`/WikiXMLConverter/src/main/java/de/mannheim/ids/wiki/WikiXMLConverter.java`
is the starting point of the conversion process. Besides,
`/WikiXMLConverter/src/main/java/de/ mannheim/ids/wiki/WikiXMLProcessor.java`
is the class managing the overall conversion process.

2.1 Properties

For article pages, a properties file requires the following properties:

- `wikidump = data/dewiki-20130728-sample.xml`
  The wikipedia dump file path.
- `language_code = de`
  Two letter language code of the wikipedia.
- `output_encoding = utf-8`
  The encoding of the output files.
- `namespace_key = 0`
  the namespace key of the wikipedia pages to convert, for instance the namespace key for articles is 0, talk pages is 1, and user talk pages is 3. See https://en.wikipedia.org/wiki/Wikipedia:Namespace.
- `max_threads = 4`
  The number of maximal threads allowed to run concurrently. Maximum is the number of CPUs -1.
- `generate_wikipage = true`
  The option to generate wikipage files in wikitext (true or false).

For talk related pages, additional properties are required as follows:

- `user_page = Benutzer`
  User page prefix in the wikidump language, for instance "*User*" in English, "*Benutzer*" in German. See https://en.wikipedia.org/wiki/Wikipedia:User_pages.
- `user_contribution = Spezial:Beiträge`
  User contribution page prefix in the wikidump language, for instance *Special:Contributions* in English, *Spezial:Beiträge* in German. See https://en.wikipedia.org/wiki/Help:User_contributions.
- `signature = Hilfe:Signatur`
  Signature page in the Wikidump language, for instance *Wikipedia:Signatures* in English, *Hilfe:Signatur* in German. See: https://en.wikipedia.org/wiki/Wikipedia:Signatures.
- `unsigned = unsigniert`
  Unsigned template in the Wikidump language, for instance *unsigned* in English, *unsigniert* in German, *non signé* in French. See https://en.wikipedia.org/wiki/Template:Unsigned.

2.2 Outputs

The output of this conversion is a WikiXML corpus, namely a collection of WikiXML files organized alphabetically and numerically by the page titles in folders. The WikiXML files are named based on the wiki page ids. Sometimes, a wikipedia page title starts with a character that cannot be normalized into an alphanumerical character. These pages are stored in Char/ folder and are ignored in the second stage conversion. The tree structure in Figure 2 illustrates how the WikiXML output files are organized.

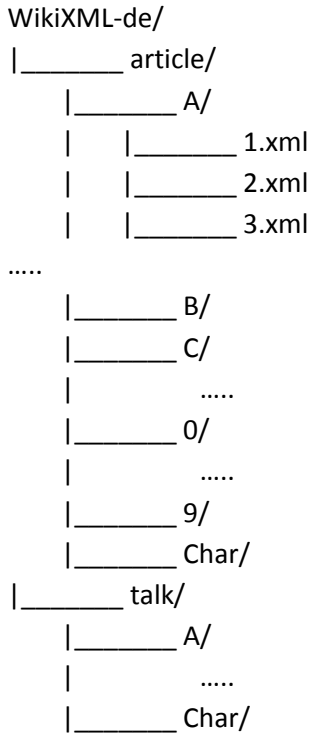

Figure 2 WikiXML Corpus Structure

2.3 Logs

The log file lists all the WikiXML files (and wikitext files) generated through the conversion. At the end of the file, the number of files and the exceptions/errors encountered during the conversion are summarized. Figure 3 shows an example of a summary from the conversion of the articles of the German wikipedia.

```
================================================
Total pages (without redirect pages) 1810405
Total non-empty pages 1809630
Total redirect pages 1256662
Total empty pages 0
Total empty parsed pages 765
Total pages without id 0
Total Sweble exceptions 0
Total Renderer exceptions 765
Total DOM exceptions 10
Total XML Page structure exceptions 0
Total thread deaths: 0
Total unknown errors: 0
================================================
WikiXMLConverter execution time 1:34:49
```

Figure 3 WikiXML Conversion Statistics of German Wikipedia Articles

Empty parsed pages shows the number of pages that become empty after the parsing process, which can be caused by Sweble exceptions or Renderer exceptions. Sweble exceptions are exceptions that are thrown by the Sweble parser and cause the parsing of the corresponding wikipage to break off. After a successful parsing resulting in an abstract syntax tree, a renderer exception might be thrown within the rendering process, namely while generating WikiXML from the abstract syntax tree.

The generated WikiXML data is not guaranteed to be well-formed. Thus, an XML well-formedness check is performed using a DOM Parser. If a generated WikiXML is not well-formed, a DOM exception will be thrown. Apart from the wikitext, which is the text content a wikipage, metadata of a wikipage is also incorporated in a WikiXML file. The page metadata structure is also checked for well-formedness.

2.4 Postings

Wikipedia talk pages contain discussions among users about the related wikipedia articles. In a typical computer-mediated communication corpus, a posting is a contribution to a written dialogue, similar to an utterance in a spoken conversation. It is a piece of text sent/posted/submitted to a server at one point of time. A collection of sequential postings within the same subject forms a conversation thread.

In wikipedia, however, postings are not necessarily sequential. Since a wiki talk page is basically a text or document, any user may write his/her post anywhere in the text. He/she may also remove or edit parts of the existing text. Thus, the boundaries between postings may be unclear. WikiXMLConverter makes use of a heuristic approach to segment wikitext and at the same time create postings from the segments (Margaretha and Lüngen, 2014).

See:
`/WikiXMLConverter/src/main/java/de/mannheim/ids/wiki/page/WikiPostHandler.java`

2.4.1 Signatures

Most postings are explicitly signed, unsigned or marked by other users. Three types of signatures are defined:

- *Signed signatures*
  are signatures that are explicitly signed by registered authors by using tildes. See https://en.wikipedia.org/wiki/Wikipedia:Signature.

- *Unsigned signatures*
  are signatures that are added by registered or unregistered users to mark an existing unsigned posting. See https://en.wikipedia.org/wiki/Template:Unsigned.

- *User contributions*
  are signatures that are added by unregistered users. For instance, when an unregistered user uses four tildes to sign his post, a special contribution link based on his ip-address will be generated instead of a user link for a registered user. In general, user contributions signify all changes made by users (https://en.wikipedia.org/wiki/Help:User_contributions).

The signature types can be specified in `<autoSignature>` elements. All the signature strings are automatic signatures because users write only signature codes in wiki markup, while the actual signature strings are automatically generated from the markup by the wiki software.

See:
`/WikiXMLConverter/src/main/java/de/mannheim/ids/wiki/page/WikiPostHandler.java`

2.4.2 Timestamps

Posting signatures usually contain information about the posting date, time and timezone. These timestamp information is useful for linguists to discover language trends, for instance to find out when a new word has appeared and started to be used widely. The time information stored in attributes or in a timeline in the metadata section cannot be used by the current corpus research software. Therefore, the timestamp is kept in the running text of the corpus, although an external XML list of timestamps are also generated.

Wiki markup for timestamps differs in wikipedias of different languages, although some languages have similar timestamp formats. In the WikiXMLConverter, different regex patterns were created to handle different timestamp formats. The patterns allow minor differences of the formats enabling them to handle some variations in the timestamps.

Wikipedia language	Timestamp format
German	11:38, 23. Jan. 2008 (CET)
Norwegian	11. feb 2008 kl. 02:27 (CET)
French	10 décembre 2007 à 12:02 (CET)
Hungarian	2006. október 17., 00:30 (CEST)

Figure 4 Various Timestamp Formats

See: `/WikiXMLConverter/src/main/java/de/mannheim/ids/wiki/page/WikiTimestamp.java`

2.5 Improper tags

Wikitext in wikipedias is written in the wiki markup language. Wiki markup defines various formats which are converted by a wiki software (MediaWiki in the case of Wikipedia) into HTML tags such as headings, text highlighting styles (e.g, italic, bold), tables, ordered and unordered lists. Besides, wikitext also contains other tags such as div, small and ref, which are written in escaped forms (i.e. using `<` and `>` to represent < and > respectively, for instance `<div class=”center”>` is written `<div class="center&quot>`). The handling of escaped tags is done in the pre-processing step (see Figure 1).

To get the tag structure within wikitexts, the escaped HTML tags must be unescaped, for instance `<` and `>` must be restored into < and > respectively. However, some of these tags are problematic, for instance sometimes some closing tags are missing and at other times, there are too many closing tags. Consequently, after unescaping the tags, the resulting HTML may not be well-formed.

Both the Sweble and TagSoup parsers can handle this problem by simply removing excessing closing tags and adding closing tags when they are missing. Nevertheless, it is unclear where the missing closing tags belong. Sweble’s and TagSoup’s strategies are similar, i.e. to greedily add a closing tag and repeatedly adding the tag pairs whenever they can until the end of a text. This behavior is, however, undesirable because it often adds many unnecessary tags and makes the structure more complex and problematic. It may also create invalid HTML structures, for example when a missing closing tag is a phrase level tag, such as `<small>`, and the added `<small>` tags surround a paragraph.

To avoid such invalid structures and reduce tag addition, unescaped wikitext is segmented per paragraph-level (separated by an empty line) and each of the wikitext segment is given to a TagSoup parser. Tag addition is thus limited within a relatively small wikitext segment and the added phrase level tags will not surround a paragraph level text.

See:
`de.mannheim.ids.wiki.page.WikiPageHandler.parseToXML(String, String, String)`
`/WikiXMLConverter/src/main/java/de/mannheim/ids/parser/TagSoupParser.java`

2.6 Problematic wikitext content

For exactly the same wikitext, Sweble sometimes throws `de.fau.cs.osr.utils.visitor.VisitingException: java.lang.StackOverflowError`, but sometimes not. For instance, Figure 5 describes a log of such a case.

```
1 SWEBLE: Discussion utilisateur:Catherine rodriguez #2875534, cause:
de.fau.cs.osr.utils.visitor.VisitingException: java.lang.StackOverflowError
''[[Image:Exemple.jpg] ]salut je mappelle catherine rodriguez  jai 12 ans je
suis au secondaire 1 jaime jouer au soccer???
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
**************************************************************************
****************************************
je vous laisse je vous  adore xxxxxx''''
```

Figure 5 StackOverflowError Example

If no exception is thrown, long sequences of asterisks are parsed into long empty nested lists.

Solution:
Heuristic approach by setting a max number of stars, in other words nested lists, and removing long stars with length more than the max number. If wikitext startswith a star, split the wikitext by space, if the second element in the stack is an empty string, then remove the whole text. Stack size can be set also larger, but it does not make sense for this case, because the wikitext is not meant to be a nested list anyway.

3 WikiXML index

The WikiXML index lists all the WikiXML files generated by the WikiXMLConventer tool. This index can be generated by using the shell script WikiXMLCorpusIndexer.sh that takes 3 arguments:

```
[Wiki page type] [WikiXML corpus folder] [output file]
```

The Wiki page type will be the root element of the index file and therefore must be in plural form, for instance articles, talks and user-talks.

Running command example:

```
./WikiXMLCorpusIndexer.sh articles WikiXML-de/article article-index.xml
```

will create an index (`article-index.xml`) of all the WikiXML files under the folder `WikiXML-de/article`. The index will have the following structure:

```
<articles>
 <index value="0">
    <id>179</id>
       ...
 </index>
 ...
 <index value="A">
    <id>1</id>
      ...
 </index>
      ...
</articles>
```

The WikiXML index is required by the WikiI5Converter to access the wiki pages and arrange them in the final I5 corpus. Thus, it is necessary to index the WikiXML files before running the WikiI5Converter.

4 Language links

Most of wikipedia articles have analogs in the other wikipedias of other languages. The links to these analog pages are not explicitly written in the wikidumps, but stored separately in a form of a database table. A wikipedia langlinks table contains information of all the page titles in different languages. The analog page titles of a wiki page are connected to each other by means of its page id.

To obtain this language link information, firstly download the corresponding language link sql dump of the wikipedia dump and restore it to a MySQL database. The database will contain a table "`langlinks`" with 3 columns:

- `ll_from` lists the wiki page ids
- `ll_lang` lists the language of the wikipedias
- `ll_title` lists the page title in different languages.

Rename the table name "`langlinks`" to "`[2-letter languagecode]_langlinks`" to distinguish the tables of wikipedias of different languages. For instance, langlinks table from a german wikipedia dump should be renamed into de_langlinks by using the following MySQL command:

```
RENAME TABLE langlinks TO de_langlinks
```

Using the langlinks table, wikipedia page titles across the wikipedias of different languages can be listed by wiki page ids. For instance, the following command:

```
SELECT * FROM de_langlinks WHERE ll_from = 3;
```

lists all the wikipedia page titles in other languages for the German wikipage of id number 3. A snapshot of the results is shown below.

```
+---------+----------+--------------------------------------------+
| ll_from | ll_lang  | ll_title                                   |
+---------+----------+--------------------------------------------+
|       3 | af       | Aktinium                                   |
|       3 | am       | አክቲኒየም                                   |
|       3 | ar       | أكتينيوم                                   |
|       3 | ast      | Actiniu                                    |
|       3 | az       | Aktinium                                   |
|       3 | be       | Актыній                                    |
```

From this table, language links to other wikipedias can be easily generated by using the following format:
`https://[ll_lang].wikipedia.org/wiki/[ll_title]`

Besides, this information is needed to create the `<relatedItem type="langlink">` in each wiki page (`<idsText>`) in a wiki I5 corpus. Thus, before running WikiI5Converter, it is necessary to first restore and rename the langlinks table that corresponds to the current wikidump.

Interaction words used as pseudo-markup

Interaction words orthographically represent speaker's actions, gestures or facial expressions (e.g. `*giggle*`) in a written conversation. They appear frequently in the conversations among the users in wikipedia talk pages and are of particular interest in the linguistic analysis of CMC. Sometimes, interaction words are even used as (pseudo) markup, as in `<seufz>nicht</seufz>` .;

I5 adopted the CMC structured proposed by Beißwenger et al. (2012) as an extension to the TEI P5 scheme. In I5, pseudo markup with interaction words is encoded as shown in Figure 6.

```
<p>Bitte
   <interactionTerm>
      <interactionWord id="WDD15.A00.00131-1-iaw1" n="1"
next="WDD15.A00.00131-2-iaw2"
topology="openingTag">&lt;seufz&gt;</interactionWord>
   </interactionTerm>
nicht
   <interactionTerm><interactionWord id="WDD15.A00.00131-2-iaw2"
n="2" prev="WDD15.A00.00131-1-iaw1"
topology="closingTag">&lt;/seufz&gt;</interactionWord>
   </interactionTerm>
 alles mit Grossbuchstaben beginnen, sondern und neue Rechtschreibung, wenn
möglich</p>
```

Figure 6 Interaction Word Markup Example

WikiI5Converter relies on a list of interaction words to recognize them in WikiXML and it only deals with interaction words that are marked as tags, for instance `<seufz>`. The lists of interaction words vary for different languages. We only provide a list of interaction words for German. The interaction words (inflectives) are contained on the file that is given as "`inflective_file`" in the properties file and must be listed as follows.

```
<inflectives>
  <name>abgreif</name>
  <name>auf-die-Nägel-blas</name>
  <name>Augenverdreh</name>
  <name>ausrück</name>
  <name>dazwischen-quetsch</name>
  <name>dazwischenquetsch</name>
  <name>Dazwischenquetsch</name>
  <name>Doppelseufz</name>
        …
</inflectives>
```

5 WikiI5Converter

The XSLT transformation in WikiI5Converter is done by using a Saxon-EE library, which is a the commercial variant of Saxon. In the development, we use saxon9ee-9.4.0.3J.jar. Both the saxon-ee jar and its license (saxon-license.lic) are needed for the conversion and should be put together in a lib/ folder. While running the WikiI5Converter tool, the `lib/` folder must be added to the java classpath. The following describes the command format to run the WikiI5Converter tool in a terminal.

```
java -Xmx4g -cp "[jar-file]:lib/*:." [main-class] -prop [properties-file-path] >
[log-file-path] 2>&1
```

The main class of the WikiI5Converter is `de.mannheim.ids.wiki.WikiI5Converter`. Similar to the WikiXMLConverter, it is advisable to set the java memory allocation pool to convert a full wikipedia and log the console output.

Command example:

```
java -Xmx4g -cp "code/WikiI5Converter-1.0.1-jar-with-dependencies.jar:lib/*:."
de.mannheim.ids.wiki.WikiI5Converter -prop properties/i5-dewiki-
article.properties > logs/wikiI5-dewiki-20150808-article.log 2>&1
```

5.1 Properties

WikiI5Converter requires the following properties in a properties file:

- `wikidump = dewiki-20130728-pages-meta-current.xml`
  The name of the wikidump is used to create the final wiki I5 corpus metadata.
- `language = Deutsch`

The language name of the wikipedia dump in its language.

- `korpusSigle = WPD13`
  The code or identifier of the corpus level.
- `namespace_key = 0`
  The namespace of the WikiXML files.
- `max_threads = 2`
  The number of maximum threads allowed to run concurrently. The maximum possible threads is the number of CPUs -1.
- `WikiXML_folder = WikiXML-de/articles`
  The location of WikiXML files to convert (root folder).
- `WikiXML_index = WikiXML-de/article-index.xml`
  The index of the WikiXML files to convert.
- `output_file = i5/dewiki-20130728-articles.i5`
  The output filename/path, namely where and how the final corpus should be named.
- `output_encoding = iso-8859-1`
  The encoding of the output file (how the final corpus should be encoded).
- `inflective_file = inflectives.xml`
  The location of the inflectives file.
- `db_url = jdbc:mysql://localhost:port/dbname`
  The URL of the database containing the corresponding langlinks table.
- `db_username = username`
  The username to connect to the database.
- `db_password = password`
  The corresponding password to the username to connect to the database.

5.2 Encoding

Wiki I5 corpora should be encoded in ISO-8859-1 for DeReKo. However, it is better to encode WikiXML corpora in UTF-8, because the original wikipedia dumps are encoded in UTF-8. While encoding WikiXML to ISO-8859-1, many UTF-8 entities not compatible with ISO-8859-1 are omitted. However, these entities are not omitted by the WikiI5Converter.

Since these entities are actually not part of ISO-8859-1, an xmllint validation of the wiki I5 corpora will not be successful. These entities should therefore be replaced with an acceptable ISO-8859-1 entity. This can be done by using perl with the following command:

```
perl -wlnpe 's/\&#xd[89a-f]..;/\&#xf8ff;/g' < [input-file] > [output-file]
```

5.3 Corpus structure

An I5 corpus has a tripartite structure, consisting of the *corpus, document* and *text* level. The root element is `<idsCorpus>` having "`korpusSigle`" as identifier, for instance WPD15 denotes German wiki I5 corpus from a dump taken in 2015. The `<idsCorpus>` element consists of many `<idsDoc>`

elements, and an <idsDoc> contain many <idsText> elements. Each of these elements has an <idsHeader> describing the contents of the corresponding elements.

Similar to the WikiXML index, each <idsDoc> is ordered by alphabets and numbers which represents the first character of the page/text titles. The documents are further grouped by the text ids, namely every 10000 ids. Document identifiers known as "documentSigle" are 1-letter and 2-digit text group ids combined with the "korpusSigle". For instance, "WPD15/A00" contains all texts with ids are between 1 to 99999, and "WPD15/A01" between 100000 to 199999. Text identifiers combine the "korpusSigle" and their corresponding "dokumentSigle" with the wikipage id. For instance, "WPD15.A00.00001" is the idsText's textSigle for the wikipage with id "1" having a title starting with A.

In some cases, for instance the English wikipedia, the length of the wiki page ids exceeds 7 digits. For these wiki pages, exceptional textSigles are generated with length 10 including the 2 digits in the text groups. For instance, "WPD15.A00.12345678" is generated for the wiki page with id "12345678".

5.4 XSLT Transformation

The XSLT transformation is only done for <idsText>. The other corpus structures and document sorting are done in Java. The main XSLT file creates the <idsHeader> for <idsText> and groups WikiXML by headings (see /WikiI5Converter/src/main/resources/Templates.xsl). It includes the additional XSLT file containing various templates handling different XML tags.

5.5 Logs

The log file lists all the WikiXML files going through the transformation process. Not all WikiXML files, however, are successfully transformed. The statistics of the transformation process is summarized in the end of the log file. The summary of the German wiki I5 conversion is shown Figure 7 below.

```
Number of transformed pages: 1802682
Number of not-transformed pages (char index): 6939
Number of transformation errors: 0
Number of empty transformation results: 0
Number of DTD validation errors: 9
Number of non well-formed XML: 0
Total number valid pages: 1802673
WikiI5Converter execution time 12:47:32
```

Figure 7 WikiI5Conversion Statistics of German Wikipedia Articles

As described in the WikiXMLIndex section, some WikiXML files whose titles cannot be normalized are put into a char/ folder, and these files are ignored for transformation. Transformation errors

are errors that are thrown by Saxon API during the XSLT transformation. After a successful transformation, the resulting `<idsText>` elements are validated against the I5 DTD (http://corpora.ids-mannheim.de/I5/DTD/i5.dtd) by using a SAX Parser. The number of DTD validation errors shows the total of invalid pages. The total number of valid pages shows the number of wiki pages in the final corpus.

Note: empty transformation results and non-wellformed XML are not used anymore and should be removed in the next version.

6 References

Beißwenger, M., Ermakova, M., Geyken, A., Lemnitzer, L., and Storrer, A. (2012). A tei schema for the representation of computer-mediated communication. Journal of the Text Encoding Initiative [Online], 3.

Margaretha, E., and Lüngen,H. (2014). Building linguistic corpora from Wikipedia articles and discussions. Journal for Language Technologie and Computational Linguistics (JLCL), 2/2014.

Lüngen, H., and Sperberg-McQueen, C. M. (2012). A TEI P5 Document Grammar for the IDS Text Model. Journal of the Text Encoding Initiative [Online], 3. URL : http://jtei.revues.org/508 ; DOI : 10.4000/jtei.508