Eliza Margaretha, Harald Lüngen

Building Linguistic Corpora from Wikipedia Articles and Discussions

Abstract

Wikipedia is a valuable resource, useful as a lingustic corpus or a dataset for many kinds of research. We built corpora from Wikipedia articles and talk pages in the I5 format, a TEI customisation used in the German Reference Corpus (Deutsches Referenzkorpus - DeReKo). Our approach is a two-stage conversion combining parsing using the Sweble parser, and transformation using XSLT stylesheets. The conversion approach is able to successfully generate rich and valid corpora regardless of languages. We also introduce a method to segment user contributions in talk pages into postings.

1 Introduction

Wikipedia is a large, multilingual and rich online encyclopedia covering a wide range of domains including medicine, sport and history in millions of articles and talk pages (discussions). As a language resource, Wikipedia is useful in multilingual natural language processing, knowledge extraction, linguistics studies, and other disciplines. Since the content of Wikipedia has not been written by a single author, but collaboratively by many users, it is also of interest in computer-mediated communication (CMC) studies.

While Wikipedia has many benefits, its content or wikitext, is rather difficult to access due to its complex structure. The structure is represented by a mixture of the *wiki markup* language[1] and HTML tags. Although there is an effort to create a wikitext standard[2] and a Wikipedia DTD[3], so far it has not been standardized. The HTML generated by the wiki software *MediaWiki*[4] (Barett, 2009) also contains structural errors (Schenkel et al., 2007; Dohrn and Riehle, 2011).

XML-based Wikipedia corpora are advantageous because Wikipedia content can be accessed by using the query language XPATH[5]. Moreover, they can easily be reused, because it does not require much effort to adapt them for other projects. Another advantage is that XML[6] can be converted into other standard formats such as (variants of) the TEI. An XML-based Wikipedia has been proved to be useful in various tasks such as semantic annotation (Atserias et al., 2008), information retrieval, and machine learning (Denoyer and Gallinari, 2006). In fact, only a few Wikipedia XML corpora are available, and commonly they only contain a small portion of Wikipedia.

[1] http://en.wikipedia.org/wiki/Help:Wiki_markup
[2] http://www.mediawiki.org/wiki/Wikitext_standard
[3] http://www.mediawiki.org/wiki/Wikipedia_DTD
[4] http://www.mediawiki.org/wiki/MediaWiki
[5] XML Path language, see http://www.w3.org/TR/xpath/
[6] Extensible Markup Language, see http://www.w3.org/TR/xml/

The aim of the work described in this paper was to implement a conversion of all German Wikipedia articles and talk pages into TEI-based corpora for integration in the *German Reference Corpus* (Deutsches Referenzkorpus - DeReKo). The conversion system should also be reusable over time and over different language versions of Wikipedia.

DeReKo is hosted by the Institut für Deutsche Sprache (IDS) in Mannheim and serves linguists as an empirical basis for research on contemporary written German. Currently it comprises more than 24 billion word tokens, distributed over many subcorpora with texts from genres as diverse as newspaper text, fiction, parliamentary debates, and specialised text (Kupietz and Lüngen, 2014). DeReKo texts are marked up for metadata and text structure according to the XML application *I5*, which is a TEI customization (Sperberg-McQueen and Lüngen, 2012) based on the Corpus Encoding Standard XCES (see Ide et al., 2000, and Section 4.1). I5 is also the internal format for data storage in the linguistic research system COSMAS II at the IDS.[7] Hence, I5 is the target format of the Wikipedia conversion described in this paper.

Our conversion approach is based on Bubenhofer et al. (2011)'s approach, who did not convert the Wikipedia wikitext directly into the IDS-XCES (the predecessor of I5) format, but divided the process into two stages: first, convert wikitext to an intermediate XML representation and second, convert the intermediate XML to IDS-XCES. The motivation behind the division was to create an intermediate corpus representing nearly all wiki markup elements used in the wikitext in XML. The WikiXML corpus is then filtered and transformed to the more constrained IDS-XCES format using XSLT[8]. Since XSLT is ideal for transforming XML into XML, it is naturally used for the second stage.

Unlike Bubenhofer et al. (2011) who used XSLT also for the first stage, we took a parsing approach using the recent Sweble parser (Dohrn and Riehle, 2011) implemented in Java. We argue that XSLT is not appropriate for the wikitext to WikiXML transformation, because the declarative nature of XSLT is not suitable for the complexity of wiki markup, which is not proper XML in the first place. On the other hand, Sweble can handle the complexity of wiki markup and generates a Java object model from wikitext. We also implemented an XML renderer to represent this object model in XML.

Beside providing the Wikipedia corpora in WikiXML and I5, our major contribution described in this paper includes the implementation of a system to convert Wikipedia articles and talk pages into a rich XML representation. For the talk pages, we introduce a posting segmentation method using delimiters and regular expressions. We also improved Bubenhofer et al. (2011)'s XSLT Stylesheets for converting WikiXML into IDS-XCES/I5.

The paper is structured as follows: In Section 2, wiki markup and the nature and structure of Wikipedia articles and talk pages are introduced. In Section 3, we present the state of the art of the development of Wikipedia corpora. In Section 4, we explain our approach to building Wikipedia corpora. We also describe the I5 target format and the posting segmentation method for Wikipedia talk pages. In Section 5, we discuss the

[7] `http://www.ids-mannheim.de/cosmas2/`

[8] Extensible Stylesheet Language, see `http://www.w3.org/TR/xslt`

Table 1: Wiki markup examples

Wiki markup	Function
`== heading level 2 ==`	heading
`----`	horizontal rule
`:indentation level 1`	indentation
`* item`	unordered list
`# item`	ordered list
`: definition 1`	definition list
`''italic text''`	italic
`'''bold text'''`	bold
`'''''bold italic text'''''`	bold italics
`<small>small text</small>`	small font-size
`O<sub>2</sub>`	subscripts
`[[target page name|link label]]`	internal link to another wiki page
`[[http://www.wikipedia.org Wikipedia]]`	external link
`[[File:Image.png]]`	image

conversion results and the evaluation of the posting segmentation. The paper ends with a conclusion in Section 6.

2 Wikipedia

2.1 Wiki Markup

Wikipedia content is not primarily written in a standard XML-based markup language such as HTML, but in a particular markup language for wikis called *wiki markup*. Text composed with this markup is called wikitext. Wikipedia uses MediaWiki as the software that runs the wiki and converts wikitext into HTML. Some examples of wiki markup are listed in Table 1. Wiki markup also includes HTML tags, for example, `<div class="center">` is used to center a block of text. Tables can be written both as HTML tables and in wiki markup format. An excerpt of wikitext describing a table in wiki markup is given in Figure 1. Some parts of a text such as quotations, poems and source code are marked separately using the `<blockquote>`, `<poem>` and `<syntaxhighlight>` tags.

Wiki markup contains *magic words*, which are special instruction words corresponding to parser functions, variables or behavior switches. For example, `{{lc:string}}` is a magic word corresponding to the parser function that converts the text "string" to lower case. Magic words of type variable are used to print the value of variables, for example `{{PAGENAME}}` will show the name of the current wiki page. The layout or behavior of a page is managed by behavior switches, for example a table of content is generated and replaces the magic word `__TOC__`.

[9] Wikitext of `http://de.wikipedia.org/wiki/Alkalimetalle` from July 27, 2013 dump

```
{| class=&quot;hintergrundfarbe2 rahmenfarbe1&quot; style=&quot;float:  right; clear:
right; margin:1em 0 1em 1em; padding:  0.1em; border-style:  solid; border-width:  1px;
empty-cells:  show&quot; | &lt;u&gt;[[Gruppe des Periodensystems|"'Gruppe"']]&lt;/u&gt;
| align=&quot;right&quot; | "'1"'
|-
| &lt;u&gt;[[Hauptgruppe|"'Hauptgruppe"']]&lt;/u&gt;
| align=&quot;right&quot; | "'1"'
|- align=&quot;center&quot;
| [[Periode des Periodensystems|"'Periode"']]
|- align=&quot;center&quot;
| [[Periode-1-Element|"'1"']]
{{Periodisches System/Element|serie=Nm|aggregat=g|protonen=1|name=Wasserstoff| symbol=H}}
|}
```

Figure 1: A table in wiki markup [9]

MediaWiki implements the notion of transclusion as a function to include some content from a *template page* into another page by using a reference in the wikitext. For example, `{{Archives}}` will include the page Template:Archives. Besides, MediaWiki allows for extending the built-in wiki markup with additional capabilities by creating *custom tags*. The `<nowiki>`, `<score>`, `<math>`, `<ref>` and `<references>` tags are examples of tag extensions for Wikipedia.

Due to the complexity of wiki markup and eventually the wikitext structure, Wikipedia content cannot be readily and easily used as a corpus. The plain text itself without all the heavy and rich structure cannot be easily accessed. We consider wikitext to be not "clean", because it combines HTML tags and wiki markup.

Although the tags in wiki markup should be contained in < and > symbols, many of the tags are "escaped" (i.e. written with < and >), for instance `&lt;small&gt;`. Some of the tags are interpreted by browsers, although many of them are not properly paired (i.e. some escaped elements lack either an opening or a closing tag). It is not always clear why they are present in a wikitext. Partly they seem to express genuine markup that is not or no longer available in HTML (like `&lt;small&gt;`, `&lt;del&gt;`, or `&lt;strike&gt;`), and partly they represent humorous pseudo markup, which will be rendered as markup on the webpage. In the discussions, for example, we found many escaped tags that are formed by interaction words (Beißwenger et al., 2012) as in `&lt;mutmaß&gt; Hängt die <i>Beaufsichtigung</i> ggf. mit der Märzrevolution zusammen ? &lt;/mutmaß&gt;`.

Overall wikitext also frequently contains ill-formed wiki markup such as the lack of a closing symbol for a table or improper line breaks, i.e. often there is no empty line between two paragraphs. These problems may lead to wrong element nesting in the generated HTML, thus creating malformed HTML. In short, considerable effort is needed to pre-process the wikitext before it can be used properly.

Figure 2: The structure of a talk page (representation adopted from Ferschke et al., 2012): a.) page title, b.) unsigned posting with insertion of IP-address, c.) signed posting d.) table of content, e.) thread heading f.) unsigned posting, g.) unstructured discussion thread, h.) discussion thread structured by indentation

2.2 Talk pages

A Wikipedia article may be associated with a *talk page* or *discussion*.[10] A talk page constitutes a piece of wikitext just like an article, i.e. wiki markup is used in it in the same way. On a talk page, users debate an article, often evaluating (parts of) its current content and arguing about whether and how it should be revised or extended, what references and images to include etc. Usually a new edit of an article is accompanied by a contribution on its talk page explaining and justifying the edit. The project described in Ferschke et al. (2012) exploits this fact to construct a corpus of discussions of the Simple English Wikipedia and provides it with dialog act annotations for research on collaborative authoring on the web.

A contribution to a Wikipedia talk page is similar to a *posting* in computer-mediated communication (CMC, see Section 3.2). A posting in CMC such as chat or discussion forums is a piece of text sent to the server by the author at a specific point in time. Postings about one particular topic typically form a thread structure (Beißwenger et al., 2012). Although a posting has a similar status as an utterance in a spoken conversation,

[10] There are also user-related "user talk pages" available, but this paper does not address them.

postings need not immediately follows each other. Together, they form a "written conversation".

In a Wikipedia talk page, a contribution ideally should be associated with its author and posting time information. In CMC corpora, it is essential to be able to distinguish the authors (in order to observe their patterns of interaction, for instance). Since a written conversation is likely to occur non-continuously, the posting time information is needed to keep track of the sequence relations in a thread. In Wikipedia talk pages, the author and posting time information are contained in the user signature. Nevertheless, users not always sign their postings, causing the boundary of a posting to become less clear or unclear.

Strictly speaking, a Wikipedia talk page contribution does not exactly correspond to a posting (as in a chat or forum communication), because in a wiki posting action, a new version of the whole wiki page is posted to the server. This means that users may edit the page in different places within one contribution. Still, a talk page is organised in dialogue structures, in which threads and sequentially ordered, posting-like dialogue turns can be identified. Following Beißwenger et al. (2012), we consider these units of dialogue as postings in our conversion.

3 Related work

3.1 Wikipedia Corpora

Although much effort has been invested in processing wikitext, not many Wikipedia corpora are publically available. Until now, only a few kinds of Wikipedia corpora with XML content have been distributed. Moreover, most of these corpora only include small selections from Wikipedia. Only a few corpora cover all articles and talk pages of a language. The corpora vary in their structures, particularly in the Wikipedia content representations and annotations.

Denoyer and Gallinari (2006) introduced an XML scheme for their Wikipedia corpus and created a multilingual corpus from Wikipedia articles in eight languages. The corpora do not cover many of the Wikipedia articles and exclude the talk pages. The largest corpus was developed for the English Wikipedia and contains about 650,000 articles, while the current English Wikipedia has more than four million articles. The resulting XML includes only a few details; nested sections, for instance, are not represented. Additional English corpora were developed and designed for use in information retrieval and machine learning purposes such as ad-hoc retrieval, categorization and clustering. The corpora were used in INEX (Initiative for the Evaluation of XML Retrieval) and WiQA (Question Answering using Wikipedia) 2006 (Jijkoun and de Rijke, 2006).

Schenkel et al. (2007) proposed YAWN, a system to automatically convert Wikipedia into an XML corpus with semantic annotations. Since the HTML tags in Wikipedia pages seem to generate malformed XML, all the HTML tags are eleminated in the pre-processing level. The elimination causes a loss of information about the text layout, which is tolerated because the focus of the corpus is the page contents. The conversion

includes section, list, tables, links, image links, and highlighting markup (i.e. bold and italic). To ensure that the corpus contains only well-formed documents, the resulting XML documents are checked for well-formedness. The corpus has a markup scheme similar to that of Denoyer and Gallinari (2006), but its structure is more detailed (for instance, nesting of sections is included). The page metadata and the page content are separated by `<header>` and `<body>` tags. Additionaly, the pages are annotated with concepts from WordNet (Fellbaum, 1998), where the concepts are identified by exploiting the lists in and the categories assigned to the pages.

The ILPS (Information and Language Processing Systems) group of the University of Amsterdam provides XML corpora based on Wikipedia [11]. The corpora were built from some portion of Wikipedia articles in different languages. The XML was particularly designed for information retrieval and natural language processing tasks in CLEF (Cross-Language Evaluation Forum). Their conversion tool is available for download.

Bubenhofer et al. (2011) built a comparable, multilingual, annotated XML corpus from all articles and discussions of the German, French, Italian, Polish, Hungarian and Norwegian Bokmål Wikipedia within the EurGr@mm project at the Institut für Deutsche Sprache in Mannheim. As mentioned in Section 1, we adapt their two-stage conversion approach and improve their XSLT Stylesheets.

Wikipedia was also used to build corpora for specific purposes, which are not necessarily formated in XML. Due to the lack of corpora for analyzing collaborative writing, Daxenberger and Gurevych (2012) built such a corpus in their study of the collaborative writing process of Wikipedia articles. First they created the Wikipedia Quality Assessment Corpus by selectively collecting featured and non-featured articles from the English Wikipedia. This corpus was used to compare the quality of the featured and non-featured articles. They also selected 891 revisions of these articles from the English Wikipedia Revision History. The revisions were compared and the difference between two adjacent revisions was defined as an edit. The number of edits was 1995, and the edits were annotated with a category, for example insert, delete or modify. Finally, the corpus contains a list of edits and its annotations.

3.2 Computer-Mediated Communication Corpora

Wikipedia discussions are an instance of computer-mediated communication (CMC). CMC exhibits peculiarities that are features of neither traditional written nor spoken communication, such as a document structure containing postings which are organised in threads or logfiles and also specific orthographic and lexical features such as the use of interaction words (e.g. inflectives **grins**) and symbols (e.g. smileys/emoticons) (Bartz et al., 2013).

The recent initiative *DeRiK - German Reference Corpus of Computer-Mediated Communication* (Beißwenger et al., 2012) has set itself the task of compiling a balanced corpus of German CMC texts to provide an empirical basis for research on German CMC language use. In the long run, DeRiK is planned to include material from all

[11] `http://ilps-vm09.science.uva.nl/WikiXML/` accessed on August 26, 2013

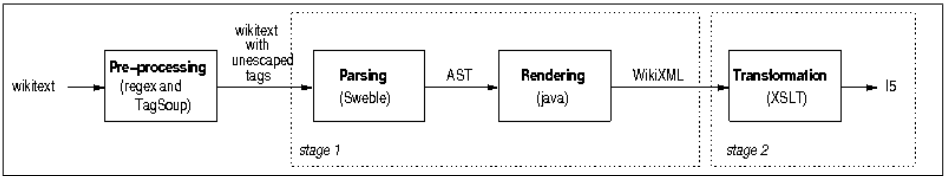

Figure 3: Wikitext to I5 conversion pipeline

major CMC genres such as email, social network communication, discussion forums, chat, Wikipedia discussion, and instant messaging.

The compilation of CMC corpora faces several obstacles. First, web content is subject to copyright restrictions, and in many cases the authors of CMC texts have never agreed that their writings be re-used in linguistic corpus projects, let alone be re-distributed among the linguistic research community, and it seems to be a hard if not impossible task to acquire such permissions retroactively. Fortunately the Wikipedia terms of use permit sharing and reusing of Wikipedia content under sufficiently free and open licenses[12].

Second, current corpus technology is in several ways not suited to the peculiarities of CMC text and CMC documents. For instance, Beißwenger et al. (2012) argue that the TEI P5 corpus encoding scheme lacks markup expressions for the specific macro and micro structure features of CMC mentioned above. To remedy the situation, they introduce a proposal for extending the TEI scheme with a module for CMC. Two encoding examples (Wikipedia talk pages and chat) according to the proposal are provided at http://www.empirikom.net/bin/view/Themen/CmcTEI.

Another CMC corpus that includes a Wikipedia subcorpus has been compiled in the the SoNaR project (van Halteren and Oostdijk, 2014).

4 Method

Our objective is to build Wikipedia corpora in the I5 format described in Section 4.1. The Wikipedia articles and talk pages are separated into two different corpora. To convert wikitext to I5, we use a pipeline architecture composed of four processing modules: pre-processing, parsing, rendering, and transformation. The pipeline is illustrated in Figure 3 and described in Section 4.3. Each Wikipedia article and talk page is processed through this pipeline.

The pipeline takes the entire wikitext of an article as an input. On the other hand, a wikitext of a talk page is segmented into postings, and the pipeline takes each posting as an input. The posting segmentation method is described in Section 4.3.2. In the first module of the pipeline, the input wikitext or posting is pre-processed by using regular expressions and TagSoup. The output of the pre-processing stage is a wikitext with

[12] CC-BY-SA (Creative Commons Attribution-ShareAlike License) and GFDLC (GNU Free Documentation License)

unescaped tags, which is given to the Sweble parser in the parsing module. The Sweble parser parses the wikitext into an abstract syntax tree (AST) that is then rendered as WikiXML. The conversion of wikitext to WikiXML is further explained in Section 4.3.1. The WikiXML content is finally transformed into I5 using XSLT stylesheets (see Section 4.3.3).

4.1 Target Markup I5

The IDS text model defines the hierarchical corpus and text structure of the German Reference Corpus (DeReKo). It was first introduced in 1992 as a home-grown, character-based format. It was recast and further extended in SGML as an adaptation of the Corpus Encoding Standard CES (Ide, 1998), and later, with the arrival of XML, in XCES (Ide et al., 2000). CES/XCES itself was based on the TEI P3 model, restricting TEI to an application to linguistic corpora. The IDS text model includes features that are not part of the TEI model, such as its tripartite corpus structure with the units *corpus - document - text*. In a corpus of literary texts, for example, each individual book constitutes a document, and each short story within a book of short stories, constitutes a text. Each of the three units has its own metadata section modeled on the teiHeader, but also including some bibliographic/metadata components that are not part of the TEI model, such as the time of creation (which may deviate considerably from the official date of publication). Besides, the text model contains elements that are present in the TEI model, but whose content model has been altered (mostly to restrict it more, but sometimes also to extend it). Because of this history, the document grammar used for the IDS text model was an adaption of the official XCES, i.e. IDS-XCES. Over the years, several new features were added when necessary (when new corpus material or analyses required it), but then care was taken that the new elements and attributes were based on the corresponding TEI specification if available.

With the advent of TEI P5 and the new ODD mechanism for TEI customisations, it became possible to specify formally how the IDS text model corresponds to the TEI and in exactly what points it deviates. Thus in 2012, a new document grammar called *I5* was introduced, specified as an ODD document which defines the IDS text model as a TEI P5 customisation (Sperberg-McQueen and Lüngen, 2012). Presently, the I5 model contains 178 elements. On the occasion of the Wikipedia conversion described in the present article, we have additionally introduced the posting structure for corpus documents as described in the proposal of (Beißwenger et al., 2012), because the available I5/TEI elements `<div>` or `<sp>` are not suitable for annotating CMC documents as also argued in Beißwenger et al. (2012). We introduced the `<posting>` element as a "divLike" element with its content model as in the TEI proposal, but we did not introduce the `<timeline>` and `<listPerson>` elements of the proposal because this information is not relevant for DeReKo, and we store it as external XML documents. Neither did we adopt the micro structure elements of the proposal because as yet we do not identify the tokens that are supposed to be annotated by the micro structure elements (i.e. inflectives, emoticons

```
[...]
<div complete="y" n="2" part="N" org="uniform" type="thread" sample="complete">
  <head type="cross">Totensonntag in der DDR</head>
  <posting indentLevel="0" who="WU00000000">
    <p>Hallo, weiß jemand ob es auch einen Totensonntag in der DDR Gab?? Danke</p>
  </posting>
  <posting indentLevel="1" synch="t00121163" who="WU00006525">
    <p>Warum sollte es den dort nicht gegeben haben? Auch in der DDR hörte das
       Kirchenjahr mit dem Ewigkeitssonntag/Totensonntag auf und das neue fing
       mit dem 1. Advent wieder an. --<autoSignature></autoSignature></p>
  </posting>
  <posting indentLevel="0" synch="t00121164" who="WU00031907">
    <p>Und weiß jemand, ob es den Totensonntag auch in Dänemark gibt??? DANKE!
      <hi rend="pt">(<hi rend="it">nicht
      <ref targOrder="u" target="de.wikipedia.org/wiki/Hilfe:Signatur">
      signierter</ref> Beitrag von</hi><autoSignature></autoSignature></hi></p>
  </posting>
</div>
[...]
```

Figure 4: Structure of discussion thread and postings according to the TEI-proposal by Beißwenger et al. (2012)

etc.). In Figure 4, we show the representation of a part of the Wikipedia talk page for *Ewigkeitssonntag*[13].

In the I5 target representation of the present Wikipedia conversion, the German Wikipedia articles collection and the German Wikipedia discussion collection each form a corpus (an XML document with the root element `<idsCorpus>`), all articles/discussions with the same initial letter of the headword form a document (element `<idsDoc>`), and each article/talk page forms a text (element `<idsText>`).

In comparison, the XML that we use for the intermediate representation between the two stages is less restricted, and no document grammar exists for it. For encoding the basic document structure, HTML tags such as `<div>` are used, but ad-hoc tag names derived from the wiki markup such as `<gallery>` also occur. Thus it mirrors more directly the features of the wiki markup, but, unlike the former, it is always well-formed.

4.2 The Sweble Parser

Dohrn and Riehle (2011) argue that the possiblities of processing wiki content are rather limited due to the fact that the wiki software running the wikis only generates HTML, which may be (and frequently is) not well-formed. The complexity of wiki markup makes it difficult for computer programs to access the wiki content. For this reason, they proposed the Sweble parser as a Java library that can handle the complexity of wiki markup and generates a machine-readable representation. The representation is an object model that can be further used to render well-formed HTML or XML. Sweble parses a wikitext using Parsing Expression Grammars (PEGS, Ford, 2004)

[13] `http://de.wikipedia.org/wiki/Diskussion:Ewigkeitssonntag`

into an object model in an abstract syntax tree (AST, Mogensen, 2011). An AST is a data structure representing the syntactic structure of a code implementation as a tree containing nodes for constants, variables, operators and statements. In Sweble, an AST is used to represent the parsing output, i.e. the objects generated from a wikitext.

Sweble takes a wikitext as input and processes it through a pipeline architecture composed of five processing steps: encoding validation, preprocessing, expansion, parsing and post-processing. In the encoding validation step, illegal characters that can harm the next processing steps are wrapped into certain entities. In the preprocessing step, redirect links, tag extensions, templates and unknown XML elements are handled and the AST nodes of the wikitext are created. Expansion is an optional step to extend the AST nodes by resolving the templates, magic words, parser functions, and tag extensions. Before starting parsing, the AST nodes are converted back to wikitext. The wikitext is subsequently analyzed by a PEG parser, and the parser generates an AST modeling the syntax and semantics of the wikitext. Finally, the post-processing step is applied to the AST. The apostrophes are interpreted and handled, the XML tags are matched, and the paragraphs are put together. The AST can be further processed by using the Visitor design pattern (Metsker, 2002), for example to generate a HTML page.

4.3 Processing Wikipedia: Parsing and Transformation Approach

4.3.1 Converting Wikitext to WikiXML

First, Wikipedia articles and talk pages are selected, and other pages, such as user, user discussion, file, and help pages, are filtered out. Since the page types correspond to their namespaces, the filtering is done by identifying the namespaces of the pages and selecting only those pages with article or talk namespaces. The redirect pages are also filtered out by identifying the redirect title in the page metadata.

Before going through the conversion pipeline, the wikitext of a Wikipedia page is separated from page metadata. The wikitext of talk pages also goes through the posting segmentation process described in Section 4.3.2. Subsequently, it becomes the input of the pre-processing stage including several tasks: unescaping tags, handling problematic symbols and correcting tags that are not well-formed.

As described in Section 2.1, wikitext contains many escaped tags. Since we intend to capture as much structure as possible, we unescape all the tags, except for the tags embedded in link markup. For example in `[[Datei:Sigmund Freud LIFE.jpg|miniatur|<center>Sigmund Freud auf einer Fotografie 1921 Aufnahme von [[Max Halberstadt]]<center>]]`, the `<center>` tags remain escaped. The left angle bracket symbols may cause problems in parsing because Sweble may falsely recognize them as a tag definition symbol, while they may alternatively be used to signify “lower than” (e.g. < 1 %). Such brackets are escaped to prevent false tag correction.

In this work, we use the Sweble parser version 2.0.0.alpha-2 for the parsing stage. The Sweble parser features a function that corrects the tags that are not properly

Table 2: Wikitext to XML conversion

Wiki Markup	XML
== Heading ==	Heading
----	<hr />
* Item	 Item
; Term	<dt>Term</dt>
: Definition	<dl>Definition</dl>
"""bold italic text"""'	bold italic text
[http://www.wikipedia.org Wikipedia]	Wikipedia

paired. However, its strategy to handle missing closing tags is rather verbose because it keeps adding closing tags until the end of the wikitext. To reduce this repetition, we use the TagSoup parser[14] (Cowan, 2002) which has a similar strategy as the Sweble parser. Instead of passing a complete wikitext, we segment the wikitext by each empty line and pass a paragraph-like segment of the wikitext as an input to TagSoup. This way, the repetition only affects a much smaller scope. Although this strategy does not ensure that the missing closing tags are placed properly, it ensures that the WikiXML is well-formed.

The wikitext output from the TagSoup parser becomes the input for the parsing stage. The Sweble parser models the wikitext into an AST. AST nodes represent wikitext elements such as paragraphs, links, and so on. Some wiki markup is not handled by Sweble, including file links.

For the rendering stage, we adapted the HTML renderer of the Sweble library and re-implemented it as an XML renderer. The XML renderer defines how the AST nodes should be expressed in WikiXML. Table 2 highlights some conversion rules from wikitext to WikiXML. Primarily, those wikitext elements which have HTML tag counterparts are converted to their HTML tag counterparts. Internal links referring to pages in Wikipedia, external links referring to sources outside Wikipedia, and image links are resolved to HTML links. Comments are removed. Templates and tag-extensions are wrapped in a tag. Other XML tags, such as <gallery> and <timeline>, are simply copied. Our WikiXML output has richer text structures than that of Schenkel et al. (2007) as it contains more tag types and Sweble parses more of the wiki markup. Like in Schenkel et al. (2007), each resulting WikiXML page containing the page metadata and the XML content is also checked for well-formedness.

4.3.2 Posting Segmentation

Considering the possible structures of posting threads in talk pages described in Section 3.2, the task of segmenting a talk page section into postings is not a trivial problem. To

[14] http://ccil.org/čowan/XML/tagsoup/

deal with this, we analyzed the structural and textual characteristics of the postings in the wikitext and created heuristic rules for segmenting them. Each posting is converted into XML by using the method described below. Moreover, the postings are annotated according to the TEI proposal for CMC corpora (Beißwenger et al., 2012). Like in the proposal, the user signature is anonymized, and the original information about the author and timestamp of a contribution is recorded in two separate XML documents. The idea behind this is to protect confidential information and to manage authorization for accessing this information.

User signature

```
Fachliche Anmerkung @Stern:  ARIS ist keine Architektur.  ARIS ist ein
Toolset zum Beschreiben von Architekturen, das ist ein Unterschied.
[[Benutzer:Ulrich.fuchs|Uli]] 14:11, 21.  Mär 2004 (CET)
```

Help signature

```
Umknicken mit der Fußsohle nach Innen bedeutet ja das
die Außenbänder beansprucht werden.  Mit logischem Denken
könnte man sich diese Frage auch selber beantworten.
&lt;small&gt;("nicht [[Hilfe:Signatur|signierter]] Beitrag von"
[[Spezial:Beiträge/212.7.177.129|212.7.177.129]] ([[Benutzer
Diskussion:212.7.177.129|Diskussion]]) 14:27, 26.  Jan.  2012 (CET))
&lt;/small&gt;
```

Unsigned Template

```
Wie werden SMD Elkos richtig gepolt?  {{unsigned|84.56.124.232|12:51, 19.
Sep.  2006 (CEST)|ALT=nicht}}
```

Horizontal line

```
Das wird glaub doch mit ß geschrieben.  Siehe dazu
[[diskussion:Rechtschreibung]]
----------
```

Heading

```
Außenband ist korrekt (ß nach langen Vokalen und Dipthongen) ss nach
kurzen Vokalen, z.B. Kuss, Fluss.  [http://www.duden.de/ Duden, Link
'Neue Rechtschreibung', Link 'Crashkurs'; Regel 4]

==&quot;Hoch/runter&quot;==
```

Figure 5: Posting delimiters [15]

For the posting segmentation, we defined a set of posting delimiters. A posting delimiter is some piece of text structure indicating the beginning or the end of a posting, thus separating one posting from another. We identified six wiki markup elements which

[15] Wikitext excerpt of `http://de.wikipedia.org/wiki/Diskussion:Architektur/Archiv`, `http://de.wikipedia.org/wiki/Diskussion:Außenbandruptur_des_oberen_Sprunggelenkes`, `http://de.wikipedia.org/wiki/Diskussion:Elektrolytkondensator/Archiv` from July 27, 2013 dump

```
Wer Mut hat, könnte vielleicht das Modell zwei der
Wikipedia-Begriffsklärung umsetzen - trau mich da nicht ran.
[[Wikipedia:Begriffsklärung]] [[Benutzer:Marc Tobias Wenzel|mTob]]
```

Posting level 1

```
:Bin mir da auch noch unsicher - wir haben noch irgendwie die
[[Baukunst]], die da mit rein muss.  Vielleicht doch besser Modell I
und unter [[Baukunst]] die &quot;eigentliche&quot; Architektur abwickeln?
[[Benutzer:Ulrich.fuchs|Uli]] 17:20, 12.  Jun 2003 (CEST)
```

Posting level 2

```
::Architektur/Baukunst gehört hier her.  Den Rest kann man unter
&quot;Architektur (Begriffserkl.)&quot; abhandeln.
```

Figure 6: Posting thread [16]

can be used as posting delimiters: user signature, help signature, unsigned template, heading, horizontal line and indentation markup. Figure 5 shows some postings in which these delimiters are used.

In Wikipedia, a user signature is an internal link to a user page followed by a posting timestamp. Ideally a posting should be signed with a user signature. A help signature is an internal link to the help page about signing a posting. When a posting is not signed, a help signature is added together with the user's IP address information and posting timestamp. Alternatively, the *unsigned* template is used to mark a posting without a signature. To detect the signatures in wikitext, we defined several regular expressions. We also defined regular expression for extracting the author and posting timestamp information from the signatures. Furthermore, the end of a posting can be identified by a horizontal line markup formed by multiple hyphen symbols, or a heading markup. Thus, any text after the last posting and before a horizontal line or heading markup is combined to form a posting.

A posting can be followed by other postings on different levels of indentation, thus creating a thread structure, which we call a posting thread. A posting thread has a smaller scope than a section thread. A section thread covers all the postings under a heading, while a posting thread covers all the postings within a list of indentations. The depth of an indentation determines the level of a posting, specified in the `@n` attribute. An example of a posting thread in wikitext is given in Figure 6. We consider each piece of wikitext with an indentation as a posting, i.e. the indentation markup is also used as a posting delimiter.

4.3.3 Converting WikiXML to I5

Each WikiXML page content is converted to I5 by means of XSLT stylesheets. We adapted the stylesheets first written by Bubenhofer et al. (2011). Firstly, the WikiXML

[16] Wikitext excerpt of `http://de.wikipedia.org/wiki/Diskussion:Architektur/Archiv` from July 27, 2013 dump

page content is split into sections by grouping the paragraph-like elements by the occurrence of headings. Since sections may have subsections, the grouping is done recursively. The level of a section is determined by the size of its heading, for example `<h2>` in XML indicates a `<div type="section" n="2">` in I5. The thread sections for postings, however, are rendered as `<div type="thread">`. Although Bubenhofer et al. (2011) implemented grouping of sections and subsections, the grouping for subsections is restricted only to subsections of the immediate sub-level. We improved the flexibity of the grouping by allowing a section to have subsections of any level. We also handle grouping of sections which are embedded in elements other than section elements.

Bubenhofer et al. (2011) simplified the conversion to IDS-XCES by including a conversion of some wiki markup to TEI elements already in their XML, such as the text highlighting elements indicating italic or bold printing. Therefore, the content of paragraphs and lists does not need to be transformed and can be simply copied in the conversion to XCES. Our approach, however, maintains a clear separation between WikiXML and I5 by using HTML tags instead of TEI elements in the WikiXML. Therefore, we substituted the corresponding TEI templates with templates handling HTML tags. We also added more templates for further WikiXML elements, which are not handled by Bubenhofer et al. (2011), such as `<dl>`, `<blockquote>` and `<caption>`.

Our WikiXML to I5 conversion involves many more transformation processes. It does not simply copy the content of paragraphs and lists, but carries out transformations for the elements inside the paragraphs and lists. First of all, the content of sections is tested for paragraph-like elements vs. phrase elements. Paragraph-like elements are elements that can appear between paragraphs, such as `<poem>`, `<list>`, and `<quote>`. Phrase elements are elements that can appear within paragraph-like elements, such as `<b>`, `<i>`,and `<sub>`. Since in I5, a section may only contain paragraph-like elements, all phrase elements are wrapped in a (new) paragraph element (`<p>`).

The `<posting>` element in WikiXML is defined based on Beißwenger et al. (2012), thus it has a structure similar to `<posting>` in I5. The author and time information are recorded in two additional XML documents. `<poem>` elements are specified in more detail in I5 with `<l>` elements designating poem line. The different kinds of lists in WikiXML are transformed into `<list>` elements.

We counted the frequencies of occurrence of tags in the intermediate WikiXML corpora and implemented specific XSLT templates for the frequently occurring elements. In doing so, we implemented context-dependent transformations for certain elements such as `<div>`. Table 3 summarizes the conversion from WikiXML to I5 for the frequently appearing tags. We delete all less frequently occurring tags (< 500 times), but not their contents. Besides, we re-generate those tags that correspond to interaction words as escaped tags in I5 because they might be of particular interest in the linguistic analysis of CMC corpora.

Table 3: WikiXML to I5 Conversion

WikiXML Tags	I5
<p>	<p>
<posting>	<posting>
<poem>	<poem> with <l> elements
, , <dl>	<list>
, <dd>, or <dt>	<item>
<blockquote>	<quote>
<caption>,	<caption>
	<gap desc="*class name*"/>
whose class appears often	
with other classes	value of the element
<pre>	or value of the element when it appears inside a paragraph
	when appears inside a <text> or a element, otherwise value of the element
<table>, <timeline>, <references>, <gallery>	<gap desc="*tag name*"/>
<span class"=*tag-extension or template*">	<gap desc="*class name*"/>
<abbr>	<abbr>
 	<lb>
<link>	<ref>
<ref>, <Ref>, <REF>	xsl:apply-templates
<b> or	<hi rend="bo">
<i>	<hi rend="it" >
<u>	<hi rend="ul">
<small> or <big>	<hi rend="pt">
<sup>	<hi rend="super">
<sub>, <tt>, , <code>, <source>	<hi rend="*tag name*">
<font>	<hi rend="*font-style*">
<syntaxhighlight>	<hi rend="syntaxhighlight">
<s>, <strike>, <del>	value of element

5 Results and Discussion

We built WikiXML article and discussion corpora for the German Wikipedia dumps from July 27, 2013, originally containing almost 2.7 million articles and about 570,000 talk pages. We removed over 1.1 million redirect pages from the set of articles and over

50 redirect pages from the set of talk pages. We also removed about 15,000 empty pages from the set of talk pages. Eventually, we parsed approximately 1.6 million articles and 0.55 million talk pages. In the parsing stage, 142 articles and 24 postings failed to get parsed. Moreover, one parsed article and two parsed talk pages were not well-formed. All the parsed and well-formed pages are successfully transformed to I5. The resulting corpora were successfully validated against the I5 DTD. The I5 file containing the articles is 16GB and contains 678 million running words. The discussions file is 4.8GB and contains 264 million running words.

5.1 Evaluation of the Posting Segmentation

By using the posting delimiters described in 4.3.2, the posting segmentation program identifies over 5.4 million postings in the German Wikipedia talk pages. To evaluate the performance of the program, we calculated precision and recall of the postings segmented by the program against posting annotations of two human annotators. Furthermore, we compare the performance of the program with a baseline segmentation using only user signatures as the posting delimiter.

Many of the talk pages are very short containing only one or two postings. To represent the variation of talk page length, we selected 49 WikiXML talk pages (7.5KB average per page, 364KB in total) randomly but with a certain distribution of long, medium long, and short pages (12 pages >10kb, 7 pages 5-10kb, 20 pages <5kb) for the evaluation dataset. We removed the <posting> tags in the WikiXML talk pages and gave them to the annotators. The annotators consulted the corresponding webpages of the talk pages, thus judging by formal as well as textual indicators where a new posting starts, and annotated the talk pages with new <posting> tags using an XML editor. The first annotator annotated in total 646 postings and the second 602 postings in the dataset.

To measure the agreement between the two annotators, we calculated Cohen's Kappa based on boundary matches. By their annotations, the annotators had categorised each of the 1024 potential boundaries (given by all paragraph-like elements contained) into either a posting/posting boundary, a posting/non-posting boundary, or a none-boundary. Based on the resulting confusion matrix, the Kappa coefficient for the two annotators is κ=0.76. This suggests that the agreement between the two annotators is fairly good. It also suggests that there is a number of postings which are ambiguous, i.e. the boundary between some postings is not obvious.

In contrast to the annotators' segmentations, the program generates 817 postings, and the baseline 499 postings. These figures suggest that the program is somewhat overly constrained and generates too many postings. One reason for this is that sometimes the usage of the wiki markup is ambiguous. For example, the indentation markup is usually used for structuring a posting thread (see Section 4.3.2), however, Wikipedia authors also use it for other purposes, such as marking a quote. On the other hand, the segmentation of the baseline is rather sparse and generates too few postings, because the baseline does not capture the unsigned postings.

Table 4: Posting segmentation performance measures

	Annotator	micro average		macro average	
		P	R	P	R
Baseline, posting-based	1	53.51	41.33	41.96	37.03
	2	42.49	35.22	34.97	32.45
Program, posting-based	1	63.40	80.19	80.61	87.86
	2	58.75	79.73	76.80	82.85
Program, boundary-based	1	75.76	96.87	85.77	97.06
	2	70.75	96.49	86.77	96.56

For the evaluation of the posting segmentation program against the two manual annotations, we provide posting-based and boundary-based precision (P) and recall (R) in comparison with the segmentations by annotators 1 and 2 in Table 4. For the posting-based measures, we used posting text to match, whereas for the boundary-based measures, we counted only boundaries that were placed between two posting segments. The micro vs. macro scopes were averaged over the set of 49 test documents described above.

The results of the first four lines in Table 4 show that the program clearly outperforms the baseline. The micro precision values are significantly lower than the recall values in all evaluation modes, reflecting the above mentioned overgeneration of segments by the program. Compared to the manual annotations, the program is able to identify about 80% of the postings marked by annotator 1 and of the postings marked by annotator 2 (cf. the micro average recalls in lines 3 and 4 Table 4).

Measures of segmentation similarity are generally based on boundary matching, not segment matching, see Inkpen, Diana and Chris Fournier (2012) for an overview. However, we additionally evaluated segment matching because there are some limitations when evaluating boundary matching.

A boundary-based evaluation is limited by the fact that at least one boundary must exist to perform a comparison. In the case where a page is judged to contain only one posting, no boundary actually exists, because the beginning and end of a file are usually not included in the set of boundaries. Our evaluation dataset, however, contains pages annotated as containing only one posting. Thus, for the boundary-based evaluation, we created one boundary after each of these postings.

Besides, boundary matching commonly yields more matches than segment matching. For instance, suppose the program generated one posting where a human annotator segmented two postings (i.e. `|xx|` vs. `|x|x|`). Then the program got 0 out of 2 postings, but still 2 out of 3 boundaries correct. Because of this, the boundary-based evaluation measures (lines 5 and 6) are generally higher than the posting-based ones (lines 3 and 4).

5.2 Discussion

We consider a parsing approach as an improvement over the use of regular expressions as in Bubenhofer et al. (2011) to handle wiki markup, because regular expressions are difficult to maintain. The Sweble parser is an on-going project that specializes in handling the complexities of wiki markup. In fact, the Sweble parser is able to handle almost all wiki markup, including templates and tag extensions. It parses a wikitext into an object model, allowing a machine to conveniently access and manipulate wikipedia content. In our case, Sweble was able to parse almost all the German Wikipedia articles and talk pages. A few pages that exhibited very complex wiki markup failed to get parsed. In the example shown in Figure 7, a complex internal link to the *Römisch-katholische Kirche* is placed in a table, which is part of the caption of the link to an image file.

```
[[Datei:Vallásos és nem hívő közösségek Magyarországon.png|miniatur|hochkant=3.0|Die
regionale Verteilung der Konfessionen nach der Volkszählung 2001:
{| class=&quot;wikitable&quot; style=&quot;margin:  auto;cellspacing=&quot;0&quot;;
font-size=80%
!  Größte Religions-&lt;br /&gt;gemeinschaft
!  [[Römisch-katholische Kirche|Römisch-katholisch]] ...  |} ]]
```

Figure 7: Wikitext excerpt failed to be parsed by Sweble [17]

Besides its complex structure, wikitext is problematic because wiki markup is often not used properly. It contains many unmatched tags and in some cases, tags do not match because the opening or closing tag is wrongly placed as for instance in `<font color=&quot;#777777&quot;>{{NaviBlock</font>..}}` where the closing `<font>` tag occurs inside a template. We attempted to improve Sweble's current strategy to fix ill-formed wiki markup by employing the TagSoup parser (Section 4.3.1), but some wiki markup is not handled by TagSoup because TagSoup only parses pure HTML or XML. From remaining ill-formed wiki markup, Sweble generates awkward XML. Figure 8 shows an example of an unmatched wiki markup tag apparently intended to print "Niger-Kongo" in bold. Since the unmatched bold tag occurs in a list, a completed bold tag is repeated until the end of the list and then also for the rest of the wikitext.

A wikitext itself may contain awkward XML behavior. For example, a phrase element may contain a paragraph element. This behavior is not allowed in I5, and thus also makes the conversion from WikiXML to I5 difficult. Figure 9 shows a list wrapped by a `<small>` tag. This kind of structure is not properly handled by Sweble, because Sweble expects a tag to be matched per line in a list. Hence, Sweble will treat the `<small>` tag as an incorrect tag.

[17] Wikipedia excerpt of `http://de.wikipedia.org/wiki/Ungarn` from July 27, 2013 dump
[18] Wikipedia excerpt of `http://de.wikipedia.org/wiki/Ayere-Ahan` from July 27, 2013 dump
[19] Wikitext excerpt of `http://de.wikipedia.org/wiki/Denkmal_zur_Geschichte_der_deutschen_Arbeiterbewegung_an_der_Gedenkstaette_Eisenacher_Parteitag_1869` from July 27, 2013 dump

Wikitext

```
** '''Niger-Kongo
** Volta-Kongo
*** Süd-Volta-Kongo
**** Benue-Kongo
***** West-Benue-Kongo
****** '''Ayere-Ahan'''
******* Ayere (3.000 Sprecher, Kwara State, Gebiet Oyi, Kabba District)
******* Ahan (300 Sprecher, Ondo State, Gebiet Ekiti, Städte Ajowa,
Igashi, Omou)
```

Sweble output

```
    <ul>
        <li> <b>Niger-Kongo</b>
            <ul><b></b>
                <li><b> Volta-Kongo</b>
                    <ul><b></b>
                        <li><b> Süd-Volta-Kongo</b>
                            <ul><b></b>
                                <li><b> Benue-Kongo</b>
                                    <ul><b></b>
                                        <li><b> West-Benue-Kongo
                                                <ul>
                                                    <li> Ayere-Ahan<b></b>
                                                        <ul><b></b>
                                                            <li><b> Ayere (3.000 Sprecher, Kwara State, Gebiet Oyi, Kabba
District)</b></li>
                                                            <li> Ahan (300 Sprecher, Ondo State, Gebiet Ekiti, Städte Ajowa, Igashi,
Omou)</li>
                                                        </ul></li>
                                                </ul></b></li>
                                    </ul></li>
                            </ul></li>
                    </ul></li>
            </ul></li>
    </ul><b>
```

Figure 8: Unmatched wiki markup [18]

Originally, the WikiXML to I5 conversion using XSLT was supposed to straightforwardly map the elements of the intermediate, unrestricted WikiXML on I5 elements, and also to filter out tags (and sometimes their content) that are not relevant in linguistic corpora in general and have no equivalent in I5. However, we eventually used the stylesheets to handle also the awkward XML structure, either because of the over-generation of tags introduced in the tag-correction process, or the wikitext behavior itself.

Our conversion system can be used for new German Wikipedia dumps in the future and also for dumps of other languages. We have tested the system by converting the French Wikipedia from Sep 4, 2013. Only a very small number of articles and postings failed to get converted or have invalid I5 (far less than 1%). However, we hope that Sweble will be further improved to deal with unstructured wiki markup esp. the problem of unmatched tags.

The treatment of the escaped XML tags in wikitext described above in the second stage is not entirely language-independent because the tags themselves are not. We have seen examples of infrequently occurring but linguistically interesting interaction words used as escaped tags in the German talk pages, but there are also German-specific frequently occurring tags such as `<div class="BoxenVerschmelzen">` occurring in the

```
&lt;small&gt; :DER EINZELNE HAT ZWEI AUGEN
:DIE PARTEI HAT TAUSEND AUGEN.
::DIE PARTEI SIEHT SIEBEN STAATEN
::DER EINZELNE SIEHT EINE STADT.
:::DER EINZELNE HAT SEINE STUNDE,
:::ABER DIE PARTEI HAT VIELE STUNDEN.
::::DER EINZELNE KANN VERNICHTET WERDEN,
::::ABER DIE PARTEI KANN NICHT VERNICHTET WERDEN.
:::::DENN SIE IST DER VORTRUPP DER MASSEN
:::::UND FÜHRT IHREN KAMPF
::::::MIT DEN METHODEN DER KLASSIKER, WELCHE GESCHÖPFT SIND
::::::AUS DER KENNTNIS DER WIRKLICHKEIT. &lt;/small&gt;
```

Figure 9: A list wrapped by a phrase element [19]

intermediate WikiXML. Our treatment of such tags is based on a derived frequency list of tags in the German Wikipedia and thus not readily applicable to the Wikipedia of other languages. Using our present tools, language-specific tags of other languages will be simply deleted and their content will be analysed for further tags by XSLT templates in the second stage of the conversion, which is a desirable behaviour as long as specific XSLT templates have not been defined for them.

Compared with other Wikipedia corpora, our I5 corpora contain richer text structures as the conversion covers more wiki markup. Unlike many corpora that only samples from Wikipedia articles, our corpora contain almost all Wikipedia articles (99.9%). Moreover, we also provide large discussion corpora from Wikipedia talk pages, including markup for postings, which have not been available before.

6 Conclusion

Our conversion system implementing the two-stage approach, i.e. first converting wikitext to unrestricted WikiXML and then filtering and transforming the WikiXML to valid I5, has proved adequate for building linguistic corpora from Wikipedia. We have shown that the resulting corpora cover almost all German Wikipedia articles and talk pages, and that the system can also be used for languages other than German. For the second conversion stage, we have improved Bubenhofer et al. (2011)'s approach and included more structures in the Wikipedia corpora.

For the discussion corpus, we have introduced a posting segmentation, the results of which highly correspond to annotations made by humans. Moreover, we have extended the document grammar I5 to accommodate thread and posting structures as occurring in Wikipedia talk pages according to the TEI proposal for CMC corpora by Beißwenger et al. (2012). Consequently, the discussion corpus can also be used in the linguistic analysis of CMC data.

The Wikipedia corpora (the article and discussion corpora described in this paper) will be available as a subcorpus in the COSMAS II corpus search and analysis system under http://www.ids-mannheim.de/cosmas2/ as of 2014. The XML corpus files (the article and discussion corpora in I5 but also in the intermediate WikiXML versions) are available for download from the Institut für Deutsche Sprache in Mannheim under the license CC-BY-SA. The I5 corpora also contain structured metadata in the TEI header format for each article and discussion page, and the running texts are also provided with markup for sentence boundaries (`<s>`) from our sentence splitter. Additionally, we offer POS tagging of the I5 corpora from the Tree Tagger (Schmid, 1994) in separate files, represented as stand-off annotations. For download and more information, see: http://www1.ids-mannheim.de/kl/projekte/korpora/verfuegbarkeit.html.

Literatur

Atserias, J., Zaragoza, H., Ciaramita, M., and Attardi, G. (2008). Semantically Annotated Snapshot of the English Wikipedia. In Calzolari, N., Choukri, K., Maegaard, B., Mariani, J., Odijk, J., Piperidis, S., and Tapias, D., editors, *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2008/.

Barett, D. J. (2009). *MediaWiki*. O'Reilly Media, Inc., USA.

Bartz, T., Beißwenger, M., and Storrer, A. (2013). Optimierung des Stuttgart-Tübingen-Tagset für die linguistische Annotation von Korpora zur internetbasierten Kommunikation: Phänomene, Herausforderungen, Erweiterungsvorschläge. *Journal for Language Technology and Computational Linguistics*, 28(1):157–198.

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.

Bubenhofer, N., Haupt, S., and Schwinn, H. (2011). A comparable Wikipedia corpus: From Wiki syntax to POS Tagged XML. In Hedeland, H., Schmidt, T., and Wörner, K., editors, *Multilingual Resources and Multilingual Applications. Proceedings of the Conference of the German Society for Computational Linguistics and Language Technology (GSCL)*, volume 96B of *Working Papers in Multilingualism*, pages 141–144. Hamburg University.

Cowan, J. (2002). TagSoup: A SAX parser in Java for nasty, ugly HTML.

Daxenberger, J. and Gurevych, I. (2012). A Corpus-Based Study of Edit Categories in Featured and Non-Featured Wikipedia Articles. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*, pages 711–726.

Denoyer, L. and Gallinari, P. (2006). The Wikipedia XML Corpus. *SIGIR Forum*, 40(1):64–69.

Dohrn, H. and Riehle, D. (2011). Design and implementation of the Sweble Wikitext parser: unlocking the structured data of Wikipedia. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration*, WikiSym '11, pages 72–81, New York, NY, USA. ACM.

Fellbaum, C., editor (1998). *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, MA ; London.

Ferschke, O., Gurevych, I., and Chebotar, Y. (2012). Behind the article: recognizing dialog acts in Wikipedia talk pages. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 777–786, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ford, B. (2004). Parsing expression grammars: a recognition-based syntactic foundation. In *ACM SIGPLAN Notices*, volume 39, pages 111–122. ACM.

Ide, N. (1998). Corpus Encoding Standard: SGML Guidelines for Encoding Linguistic Corpora. In *Proceedings of the First International Language Resources and Evaluation Conference (LREC)*, page 463–470, Grananda, Spain.

Ide, N., Bonhomme, P., and Romary, L. (2000). XCES: An XML-based standard for linguistic corpora. In *Proceedings of the Second Language Resources and Evaluation Conference (LREC)*, page 825–830, Athens, Greece.

Inkpen, Diana and Chris Fournier (2012). Segmentation similarity and agreement. In *Proceedings of Human Language Technologies: The 2012 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 152–161, Stroudsberg, PA.

Jijkoun, V. and de Rijke, M. (2006). Overview of the WiQA Task at CLEF 2006. In *CLEF*, pages 265–274.

Kupietz, M. and Lüngen, H. (2014). Recent Developments in DeReKo. In Calzolari, N., Choukri, K., Declerck, T., Loftsson, H., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Metsker, S. J. (2002). *Design Patterns Java Workbook*. Pearson Education, Inc., US.

Mogensen, T. A. (2011). *Introduction to Compiler Design*. Springer, London.

Schenkel, R., Suchanek, F. M., and Kasneci, G. (2007). YAWN: A Semantically Annotated Wikipedia XML Corpus. In *12th GI Conference on Databases in Business, Technology and Web (BTW 2007)*, Aachen,Germany.

Schmid, H. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, UK.

Sperberg-McQueen, C. and Lüngen, H. (2012). A TEI P5 Document Grammar for the IDS Text Model. *Journal of the Text Encoding Initiative (jTEI)*, 3.

van Halteren, H. and Oostdijk, N. (2014). Variability in Dutch Tweets. An estimate of the proportion of deviant word tokens. *Journal for Language Technology and Computational Linguistics*, 29.