A Large-Scale Web Data Collection as a Natural Language Processing Infrastructure

Keiji Shinzato† **Daisuke Kawahara**‡ **Chikara Hashimoto**‡‡ **Sadao Kurohashi**†

†Kyoto University
Yoshida-honmachi, Sakyo-ku, Kyoto, 606-8501, Japan
{shinzato, kuro}@nlp.kuee.kyoto-u.ac.jp
‡National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0289 Japan
dk@nict.go.jp
‡‡ Yamagata University
4-3-16 Jonan, Yonezawa-shi Yamagata, 992-8510 Japan
ch@yamagata.ac.jp

Abstract

In recent years, language resources acquired from the Web are released, and these data improve the performance of applications in several NLP tasks. Although the language resources based on the web page unit are useful in NLP tasks and applications such as knowledge acquisition, document retrieval and document summarization, such language resources are not released so far. In this paper, we propose a data format for results of web page processing, and a search engine infrastructure which makes it possible to share approximately 100 million Japanese web data. By obtaining the web data, NLP researchers are enabled to begin their own processing immediately without analyzing web pages by themselves.

1. Introduction

Language resources such as corpora and lexicons have contributed to boost up the development of Natural Language Processing (NLP) technology. It is important to share the language resources. In recent years, language resources acquired from the Web such as 5-gram word sequences (Brants and Franz, 2006) and 0.5 billions sentences (Kawahara and Kurohashi, 2006) are released, and these data improve the performance of applications in several NLP tasks (Och, 2006).

The current language resources acquired from the Web are mainly data based on the word unit and the sentence unit. These data are useful to estimate statistical information such as co-occurrence frequencies between words. On the other hand, language resources based on the document unit are not released so far. The language resources based on the document unit are useful in NLP tasks and applications such as knowledge acquisition, document retrieval and document summarization. The performance of these tasks heavily depends on the amount of text data, and these applications mainly aim at web pages. Hence, a large amount of analyzed web pages is desirable as the language resources based on the document unit.

In this paper, we propose a data format for results of web page processing, and a search engine infrastructure which makes it possible to share approximately 100 million Japanese web data. We call the format Web standard format. A tagset is defined in the Web standard format by XML language, and is utilized for annotating commonly required analysis data in the web page processing such as results of sentence boundary detection, morphological analysis and parsing.

The annotated data is accessible via the search engine infrastructure. By obtaining the annotated data, NLP researchers are enabled to utilize web data in their own processing immediately without analyzing web pages by themselves. The infrastructure also provides the capability of retrieving web data according to several constraints. This allows the users to efficiently collect web data that the users want.

This paper is organized as follows. Section 2 describes the tagset defined in the Web standard format, and a procedure for converting web pages into web standard format data. Section 3 shows a web data collection which is constructed from 100 million Japanese web pages. Section 4 presents an search engine infrastructure for providing web data.

2. Web Standard Format

2.1. Tagset

The web standard format is a simple XML data format in which commonly required data in web page processing can be annotated. A tagset defined in the Web standard format is shown in Table 1. We classified the commonly required data into meta data and text data of a web page. The meta data and text data are annotated using the tagset. An example of annotated data is shown in Figure 1. An annotated data and its corresponding web page are assigned the same nine-digit ID for efficiently providing them via the search engine infrastructure described in Section 4.

Web data annotated in the Web standard format tagset always begins with the `<StandardFormat>` tag. The `<StandardFormat>` tag has `URL`, `OriginalEncoding` and `Time` attributes, and values of these attributes correspond to a URL, original character encoding and a crawled date of a web page respectively. The remaining meta data and text data are annotated by `<Header>` and `<Text>` tags respectively.

Table 1: The tagset defined in the Web standard format.

Tag	Description
StandardFormat	The whole data. A standard format data must be one `<StandardFormat>` element. **Attribute:** **Url:** a URL of a web page. **OriginalEncoding:** A character encoding of a web page. **Time:** A crawled data. **Child element:** Header, Text
Header	Meta data of a web page. **Child element:** Title, InLinks, OutLinks, Keywords, Description
Title	A title of a web page. **Attribute:** **Offset:** A byte offset length from the beginning of a web page. **Length:** A byte length of a title string. **Id:** A sentence ID. **Child element:** RawString, Annotation
InLinks	A set of in-links. **Child element:** InLink
InLink	An in-link. **Child element:** RawString, Annotation, DocIDs
OutLinks	A set of out-links. **Child element:** OutLink
OutLink	An out-link. **Child element:** RawString, Annotation, DocIDs

Tag	Description
DocIDs	A set of page IDs. **Child element:** DocID
DocID	A page ID. **Child element:** A page ID (a nine-digit number).
Keywords	A keywords attribute value in a META tag. **Child element:** RawString, Annotation
Description	A description attribute value in a META tag. **Child element:** RawString, Annotation
Text	Text data of a web page. **Child element:** S
S	A sentence in a web page. **Attribute:** **Offset:** A byte offset length from the beginning of a web page. **Length:** A byte length of a title string. **Id:** A sentence ID. **Child element:** RawString, Annotation
RawString	A string extracted as a sentence. **Child element:** A string extracted as a sentence.
Annotation	A result of analyzing a sentence by an NLP tool. **Attribute:** **Scheme:** A name of a tool. **Child element:** Output of an NLP tool.

These tags are included in the `<StandardFormat>` tag as its child elements.

While the `<Header>` tag includes a title, inlinks, outlinks, keywords and descriptions of a web page, the `<Text>` tag includes sentences and results of analyzing the sentences by existing NLP tools. Each extracted sentence is represented by an `<S>` tag. A byte offset and length of a sentence are annotated as attributes of an `<S>` tag. A string extracted as a sentence is enclosed by `<RawString>` tags, and its analyzed result is enclosed by `<Annotation>` tags. If the sentence is analyzed by multiple NLP tools, multiple `<Annotation>` tags with a different `Scheme` attribute value can be annotated to the identical web data.

The sentences in a web page and their analyzed results can be obtained by looking at `<RawString>` and `<Annotation>` tags in the standard format data. This allows NLP researchers to utilize web data in their own processing immediately without analyzing web pages by themselves.

2.2. Web Standard Format Conversion

The conversion procedure consists of the following four steps:

Step 1: Extract sentences from a Japaneses web page based on HTML tags and surface information.

Step 2: Conduct existing NLP tools for the extracted sentences.

Step 3: Create standard format data from crawler's log data, the extracted sentences and the analyzed results.

Step 4: Assign the same page ID to a web standard format data and its corresponding web page.

We assumed that the URL and crawled date of a web page are included in crawler's log data in Step 3.

In the conversion process, the most important procedure is sentence extraction. It is necessary for extracting sentences to detect sentence boundaries in a web page text. HTML tags and surface information are used as clues for sentence boundary detection. More precisely, block-level elements, such as heading tags `<h1>`, paragraph tags`<p>` and table tags`<table>` are used as clues. As surface information clues, a punctuation (。) and punctuation-like symbols (？, ！, ♪ and …) are used. In addition, face marks such as （＊^^＊） and （^^;） are also used. Face marks often appear in many web and blog pages, and are likely to be put in the end of sentences. Face marks are useful to detect sentence boundaries in a text which does not include punctuations.

```
<StandardFormat Url="http://www.kantei.go.jp/jp/koizumiprofile/1_sinnen.html" Original
Encoding="shiftjis" Time="2007-06-28 09:10:00">
<Header>
  <Title Offset="21" Length="39" Id="0">
    <RawString>小泉総理プロフィール・信念</RawString>
  </Title>
  <OutLinks>
    <OutLink>
      <RawString>トップ</RawString>
      <DocIDs>
        <DocID Url="www.kantei.go.jp/index.html">060936437</DocID>
      </DocIDs>
    </OutLink>
  </OutLinks>
  <InLinks>
    <InLink>
      <RawString>小泉総理の信念</RawString>
      <DocIDs>
        <DocID Url="http://mocuromi365.yh.land.to/">067985366</DocID>
      </DocIDs>
    </InLink>
  </InLinks>
</Header>
<Text>
<S Id="1" Length="70" Offset="525">
  <RawString>
    小泉総理の好きな格言のひとつに「無信不立(信無くば立たず)」があります．
  </RawString>
  <Annotation Scheme="KNP">
    <![CDATA[
      * 1D <文頭><サ変><人名><助詞><連体修飾><体言><係:ノ格><区切:0-4><RID:1056>
      小泉 こいずみ 小泉 名詞 6 人名 5 * 0 * 0 NIL <文頭><漢字><かな漢字><名詞相当語><自立><タグ単位
      始><文節始><固有キー>
… (snip) …
      ます ます ます 接尾辞 14 動詞性接尾辞 7 動詞性接尾辞ます型 31 基本形 2 NIL <表現文末><かな漢字>
      <ひらがな><活用語><付属><非独立無意味接尾辞>
      ． ． ． 特殊 1 句点 1 * 0 * 0 NIL <文末><英記号><記号><付属>
    EOS]]>
  </Annotation>
</S>
<S Id="2" Length="160" Offset="595">
  <RawString>
    論語の下篇「顔淵」の言葉で，弟子の子貢（しこう）が政治について尋ねたところ，孔子は「食料を十分にし軍備を
    十分にして，人民には信頼を持たせることだ」と答えました．
  </RawString>
  <Annotation Scheme="KNP">
    <![CDATA[
      * 1D <文頭><助詞><連体修飾><体言><係:ノ格><区切:0-4><RID:1056>
      論 ろん 論 名詞 6 普通名詞 1 * 0 * 0 "漢字読み:音 代表表記:論" <漢字読み:音><代表表記:論><文頭>
      <漢字><かな漢字><名詞相当語><自立><タグ単位始><文節始>
… (snip) …
      ました ました ます 接尾辞 14 動詞性接尾辞 7 動詞性接尾辞ます型 31 タ形 5 NIL <表現文末><かな漢字>
      <ひらがな><活用語><付属><非独立無意味接尾辞>
      ． ． ． 特殊 1 句点 1 * 0 * 0 NIL <文末><英記号><記号><付属>
    EOS]]>
  </Annotation>
</S>
… (snip) …
</Text>
</StandardFormat>
```


Figure 1: An example of the Web standard format data. (inc. parsing results of sentences)

3. Web Data Collection Construction

We have crawled web pages, and converted them into web standard format data. As a result, web data collection consisting of approximately 100 million web standard format data was constructed. In this part, we describe this collection.

3.1. Web Page Crawling

We have crawled 233 million web pages over three months, May - July in 2007, by using the Shim-Crawler[1]. The crawled web pages consist of pages written not only in Japanese but also in other languages. The crawler has been run on ten cluster machines in parallel. Each cluster machine consists of 1 CPU, 4 GB main memory and 400 GB local storage.

We use the Shim-Crawler's log data including URLs and crawl dates of web pages for constructing the web standard format data.

3.2. Web Data Collection Construction

After crawling web pages, we have converted the crawled pages into web standard format data. Although the Web standard format does not depend on language, we have converted only Japanese web pages into web standard format data.

We have first selected Japanese web pages out of the crawled pages. We used a method proposed in (Kawahara and Kurohashi, 2006) for Japanese web page selection. Briefly speaking, this Japanese page selection procedure uses character encoding information and linguistic information of a web page as clues for selection. The Japanese page selection procedure, first, roughly extracts Japanese web page candidates by looking at character encodings of web pages, and then picks up Japanese web pages from the extracted candidates according to the ratio of Japanese postpositions.

We performed the Japanese page selection procedure on the 233 million web pages. As a result of the selection, 100 million web pages were obtained. In other words, the remaining 133 million web pages were regarded as non-Japanese pages by the Japanese web page selection procedure based on character encoding and linguistic information.

After selecting Japanese web pages, we have converted these web pages into Web standard format data through the conversion procedure described in Section 2.2. As results of existing NLP tools, we added the results of the Japanese parser KNP (Kurohashi and Nagao, 1994). In the conversion process, the Japanese web pages are organized into 10,000 page sets (i.e., one page set consists of 10,000 web pages.) The page sets were processed by 162 cluster machines in parallel. Each cluster machine consists of 4 CPU cores and 4 GB main memory. To submit these jobs to the cluster machines, we used a grid shell GXP2 (Kaneda et al., 2002). It took two weeks to finish the conversion.

The comparison between original web pages and the standard format data in terms of file size is shown in Table 2.

[1]http://www.logos.t.u-tokyo.ac.jp/crawler/

Table 2: File size comparison between original web pages and web standard format data (The number of web pages is 100 millions, and web pages and web standard format data are compressed by gzip.).

Type	File size [TB]
Web pages	0.6
Web standard format data	3.1

We can see that the file size of the web standard format data is over five times bigger than that of the original web pages.

4. Search Engine Infrastructure for Providing Web Standard Format Data

In this section, we describe the search engine infrastructure for providing the web standard format data described in Section 3. Users of the infrastructure are enabled to efficiently obtain web data that match several constraints. The infrastructure is accessible via a web Application Programming Interface (API).

4.1. Search Engine TSUBAKI

The crawled web pages described in Section 3. are indexed in TSUBAKI (Shinzato et al., 2008) which is a platform to help the development of new information access methodology. TSUBAKI enables us to retrieve web pages that have relevance to any queries from 100 million Japanese web pages.

TSUBAKI provides its API without any restrictions such as the limited number of API calls a day and the number of results returned from an API per query which are typical restrictions of the existing search engine APIs. TSUBAKI API can be queried by "REST (Fielding, 2000)-Like" operators in the same way as Yahoo! API. The API users can obtain search results through HTTP requests with URL-encoded parameters. Examples of the available request parameters are listed in Table 3.

4.2. How to Obtain Web Standard Format Data

The procedure for obtaining web standard format data consists of two steps:

Step 1: Collect Web Page IDs A crawled web page and web standard format data corresponding to the page are assigned the same page ID in TSUBAKI. As the first step to obtain web standard format data, our infrastructure users have to collect web page IDs by requesting a query to TSUBAKI API. For instance, the request to obtain the search result ranked at top 20 for the search query "京都 (*Kyoto*)" is below.

http://tsubaki.ixnlp.nii.ac.jp/api.cgi?query=%E4%BA%AC%E9%83%BD&starts=1&results=20

The API returns the XML document shown in Figure 2 for the above request. The `<Result>` tag corresponds to a web page that matches the given query, and they are sorted according to the ranking measure in TSUBAKI. We

```
<ResultSet time="2008-04-02 10:48:55" query="京都" totalResultsAvailable="4867551" totalR
esultsReturned="20" firstResultPosition="1" logicalOperator="AND" forceDpnd="0" dpnd="1
" filterSimpages="1" sort_by="score">
  <Result Rank="1" Id="017307147" Score="8.87700">
    <Title>
      ＪＴＢ　ｅ－Ｈｏｔｅｌの京都府のホテル・旅館一覧
    </Title>
    <Url>http://www.docch.net/blog/jtb-e/kyouto.shtml</Url>
    <Cache>
      <Url>
        http://tsubaki.ixnlp.nii.ac.jp/index.cgi?cache=017307147&KEYS=%E4%BA%AC%E9%83%BD
      </Url>
      <Size>2900</Size>
    </Cache>
  </Result>
  <Result Rank="2" Id="046817588" Score="8.87415">
    <Title>
      京都府１の激安ホテル、格安旅館（Ｙａｈｏｏ！トラベル、ＪＴＢ、ｅ－ｈｏｔｅｌ、じゃらん、ゆこゆこネット、
      ぐるなびトラベル）
    </Title>
    <Url>http://www.goodsite7.com/hotel/25kyoto1.html</Url>
    <Cache>
      <Url>
        http://tsubaki.ixnlp.nii.ac.jp/index.cgi?cache=046817588&KEYS=%E4%BA%AC%E9%83%BD
      </Url>
      <Size>9759</Size>
    </Cache>
  </Result>
 ··· (snip) ···
</ResultSet>
```


Figure 2: An example of a search result returned from TSUBAKI API.

can collect page IDs by looking at the ID attribute in each `<Result>` tag.

In addition, the title and URL of a web page are also included in the search result. Hence, if users want to obtain the title and URL of a web page only, they do not need to proceed to Step 2.

Step 2: Obtain Web Standard Format Data using Page IDs The page IDs in a search result enable the users to obtain web standard format data. An example request for obtaining the web standard format data with page ID 01234567 is shown below.

http://tsubaki.ixnlp.nii.ac.jp/api.cgi?id=01234567&format =xml

As a result, the web standard format data of page ID 01234567 is returned from TSUBAKI API. If users want to obtain an original web page of ID 01234567, users have to change the value of "format" in the above request into "html".

5. Conclusion

In this paper, we have proposed the data format for web data, and also proposed a search engine infrastructure which makes it possible to share large-scale web data via Web API. The proposed infrastructure allows the users to obtain 100 million Japanese web pages and their analyzed results by requesting a query to the Web API.

6. References

Thorsten Brants and Alex Franz, 2006. *Web 1T 5-gram version 1*. LDC2006T13.

Roy Thomas Fielding. 2000. *Architectural Styles and the Design of Network-based Software Architectures*. Ph.D. thesis, University of California, Irvine.

Kenji Kaneda, Kenjiro Taura, and Akinori Yonezawa. 2002. Virtual private grid: A command shell for utilizing hundreds of machines efficiently. In *In 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2002)*.

Daisuke Kawahara and Sadao Kurohashi. 2006. Case frame compilation from the web using high-performance computing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC2006)*, pages 1344–1347.

Sadao Kurohashi and Makoto Nagao. 1994. A syntactic analysis method of long japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, 20(4):507–534.

Franz Josef Och. 2006. Oral presentation: The google machine translation system. In *NIST 2006 Machine Translation Workshop*.

Keiji Shinzato, Tomohide Shibata, Daisuke Kawahara, Chikara Hashimoto, and Sadao Kurohashi. 2008. Tsubaki: An open search engine infrastructure for developing new information access methodology. In *Proceedings of Third International Joint Conference on Natural*

Table 3: The request parameters of TSUBAKI API.

Parameter	Value	Description
query	*string*	The query to search for (UTF-8 encoded). The query parameter is required for obtaining search results.
start	*integer*: default 1	The starting result position to return.
results	*integer*: default 20	The number of results to return.
logical_operator	AND/OR: default AND	The logical operation to search for.
snippets	0/1: default 0	Set to 1 to obtain snippets.
filter_simpages	0/1: default 0	Specifies whether to allow multiple results with similar content. Enter a 1 to discard similar content.
id	*string*	The document ID to obtain a cached web page or standard format data corresponding to the ID. This parameter is required for obtaining web pages or standard format data.
format	html/xml	The document type to return. This parameter is required if the parameter id is set.

Language Processing (IJCNLP2008), pages 189–196.