Extracting domain knowledge for dialogue systems from unstructured Web pages

Masahiro Araki† Masaki Fujisawa† Takuya Nishimoto†
Yasuhisa Niimi†

†Department of Electronics and Information Science
Kyoto Institute of Technology
Matsugasaki Sakyo-ku Kyoto 606-8585, Japan
{araki,fujisawa,nishi,niimi}@dj.kit.ac.jp
Tel: +81-75-724-7473
Fax: +81-75-724-7400

Keywords: information extraction, dialogue system, HTML, XML

Abstract

We propose a semi-automatic domain knowledge extraction method from unstructured Web pages written in HTML. It translates Web pages to structured XML documents which can be used as a domain knowledge of task-oriented dialogue systems. In this paper, we explain an outline of our project and report preliminary results in a restricted task domain.

1 Introduction

The World Wide Web contains a wealth of information. Many people get information from the Web using GUI (Graphical User Interface) based browsers. If we could access web content via voice channels, we could get such information anytime and anywhere.

However, the simple replacement of a mouse click to word recognizer has failed to realize efficient and satisfactory information seeking dialogue. The dialogue tends to be long and irritating. The merits of using spoken dialogue are direct access of target information, stepwise clarification and collaborative responses. Task-oriented spoken dialogue systems should achieve some of these functions.

If we can translate Web contents, which are written in HTML (Hyper Text Markup Language), to the knowledge representation for such collaborative dialogue systems, we can achieve good computer and telephony integration [Sorderland 1997]. For this purpose, XML (eXtensible Markup Language) is one of the most promising languages for knowledge representation and information exchange. Therefore, we set our research goal to make a converter from HTML documents to XML.

Section 2 describes the difficulty of extracting structured information from present Web pages. Section 3 explains our conversion algorithm. Section 4 reports the results of our conversion experiments. Section 5 explains how to use this knowledge for spoken dialogue systems. Section 6 includes a conclusion and our future projects.

2 Analysis of Web pages

In order to convert from HTML to XML, we have to extract structural information from HTML documents. However, many Web pages use HTML tags for the purpose of display layout (*e.g.* the definition list tag <dl >is always used for as a simple indentation).

Also, because some types of HTML close tag (*e.g.* , </p >, etc.) can be omitted, it is difficult to properly segment HTML documents. Therefore, it is hard to make a proper content tree (or a hierarchical information structure) from HTML documents.

Figure 1 shows an example of a part of a hotel information page written in HTML.

```
<center>
<h2> ABC HOTEL </h2>
</center>
<div align="left"> Room Types </div>
<dl>
  <dd> Single
  <dd> Twin
  <dd> Double
</dl>
<hr>
<b> TEL </b><br> 012-3456-7890 <br>
```

Figure 1: Example of hotel information page (HTML).

If we want to use such documents as a knowledge base for dialogue systems, we have to extract elementary information and assign it to the proper conceptual slot.

An XML representation (Figure 2) is one of its solutions. In XML, an information structure is given in DTD (Figure 3). Therefore, we can regard XML documents as a knowledge representation for various uses.

```
<hotel>
  <name>ABC Hotel</name>
  <accommodation>
    <roomtype>
      Single
    </roomtype>
    <roomtype>
      Twin
    </roomtype>
    <roomtype>
      Double
    </roomtype>
  </accommodation>
  <phone>012-3456-7890</phone>
</hotel>
```

Figure 2: Example of hotel information page (XML).

```
<!ELEMENT hotel (name | accommodation |
                 phone | fax | address) >
<!ELEMENT name (#PCDATA) >
<!ELEMENT accommodation (roomtype)* >
<!ELEMENT phone (#PCDATA) >
<!ELEMENT fax (#PCDATA) >
<!ELEMENT address (#PCDATA) >
<!ELEMENT roomtype (#PCDATA) >
```

Figure 3: Data Type Definition of hotel information.

3 HTML-to-XML Converter

Considering the structural difficulty of HTML documents, our conversion method requires a two step procedure: segmentation and assignment.

In the segmentation step, the tags which do not contribute to the information structure (*e.g.* <font ...>, <img ... >, etc.) are eliminated and the places of the close tags are guessed based on the segmentation break rules. Then, each content enclosed by tags is extracted, making a minimal chunk of information. After that, we use a hand-coded, task-dependent thesaurus to merge related neighboring chunks to make a proper information chunk (*e.g.* hotel address, room types, etc.) Figure 4 shows a segmentation process of Web pages.

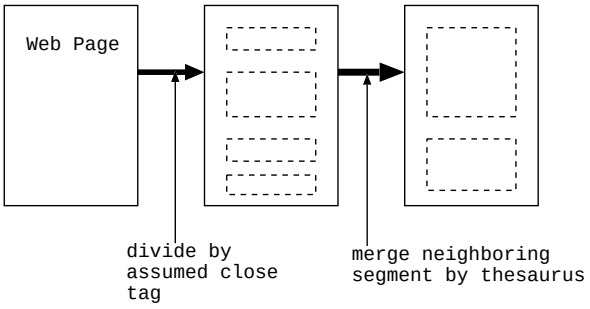

Figure 4: Segmentation process of Web pages.

In the assignment step, we also use the hand-coded thesaurus in order to determine which information chunk fits the content of the XML leaf tags. For example, in the hotel information domain, the name of the leaf tags are hotel name, address, phone, fax, transportation facilities, room type, fare, and check in-out time. Figure 5 shows an assignment process to XML contents.

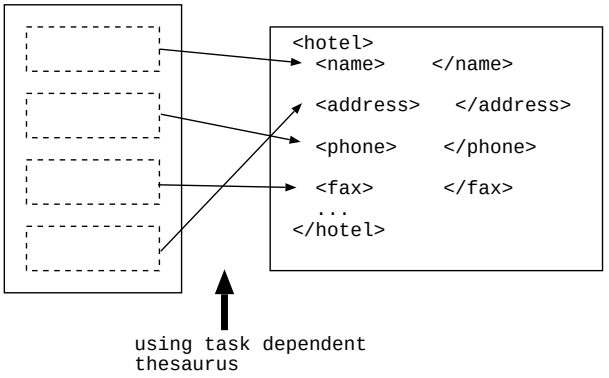

Figure 5: Assignment process to XML contents.

4 Experimental results

We did a preliminary experiment in the hotel information domain and ski resort information domain. The results are shown in Table 1. We examined 10 hotel sites and 20 ski resort sites. All pages were written in Japanese.

We define the result as proper if necessary and enough information for the slot is acquired. Also we define the result as correct if it includes some superfluous information. For example, in address tag, the proper case is just the address of the hotel and correct case includes additional information such as '(10 minutes walk from Kyoto station.)'.

From the hotel information pages, we obtained 36% proper information. Also, 31% contained correct information. The later cases can be modified by a post-processing program according to the contents (*e.g.* phone number sequence checker).

For ski resort information, we got 26% proper and 38% correct information respectively. We used a task-dependent thesaurus and a task independent extraction procedure. If we add more task dependent extraction rules, we could improve performance.

Table 1: Experimental results of HTML-to-XML conversion.

Task	proper	correct	proper+correct	failed
Hotel information	36.0%	30.7%	66.7%	33.3%
Ski resort information	26.2%	37.5%	63.7%	36.3%

5 Using XML for spoken dialogue systems

In some typical patterns of XML's DTD (Document Type Description), we found that they were suitable for use as a knowledge base for spoken dialogue systems. Such typical patterns can be translated into VoiceXML (Voice eXtensible Markup Language) [VoiceXMLForum 2000] using a dialogue library [Araki et al. 1999].

VoiceXML is expected to be a new standard used when making the Internet content and information accessible via voice and phone. Mixed-initiative, cooperative dialogue can be realized by specifying the pattern of dialogue in VoiceXML.

Therefore, our next goal is to make a converter from XML documents to VoiceXML. The outline of our system is shown in Figure 6.

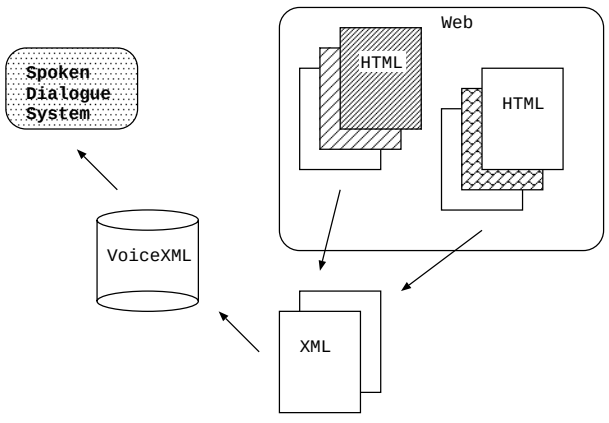

Figure 6: Contents acquisition from the Web.

Acquired XML files can be regarded as regulated like a tree structured document. Therefore, the root VoiceXML file can be constructed by prompting top level contents and the utterance "How may I help you?". If the user specifies one of its contents, the system jumps to the corresponding VoiceXML file and continues dialogue. Alternatively, if the user replies with a help type utterance, the system lists up all the top level headings. Figure 7 shows an example of a top VoiceXML file in a hotel information task.

For each specific content, if the information is an enumerated pattern, it can be translated as successive explanations. On the other hand, if the contents are itemized patterns, the explanation is of the select-and-go type. The former example is giving transportation information (Figure 8) and the latter is room type information (Figure 9).

6 Conclusion

This paper described the aim and current status of our project. We have implemented a preliminary version of HTML-to-XML converter based on a robust document segmentation technique and task domain thesaurus. We plan to evaluate the validity of the converter by examining various types of tasks.

References

[Sorderland 1997] S. Sorderland. *Learning to extract text-based information from the world wide web.* In *Proc. 3rd International Conference on Knowledge Discovery and Data Mining*, 1997.

```
<?xml version="1.0"?>
<vxml version="1.0">
 <form>
  <block> This is ABC hotel information. </block>
  <field name="item">
   <prompt> How may I help you? </prompt>
   <grammar src="top.gram"/>
   <help> The menu item is room type, address,
          phone, fax and transportation. </help>
  </field>
  <if cond="item=room type">
    <goto next="roomtype.vxml"/>
  </if>
  <if cond="item=address">
    <goto next="address.vxml"/>
  </if>
  <if cond="item=phone">
    <goto next="phone.vxml"/>
  </if>
  <if cond="item=fax">
    <goto next="fax.vxml"/>
  </if>
  <if cond="item=transportation">
    <goto next="transportation.vxml"/>
  </if>
 </form>
</vxml>
```

Figure 7: Root document in a hotel information system.

```
<?xml version="1.0"?>
<vxml version="1.0">
 <form id="0">
  <block>
    <prompt timeout="3s">
      From Kyoto station, take a city bus service line 206.
    </prompt>
  </block>
  <noinput>
    <field name="confirm" type="boolean">
    <prompt> Do you get it? </prompt>
    <if cond="confirm=false"> <goto next="operator.vxml"> </if>
  </noinput>
  <goto next="#1">
 </form>

 <form id="1">
  <block>
    <prompt timeout="3s">
      And get off the bus at ABC street. It takes about
      10 minutes. The hotel is at the other side of the
      bus stop.
    </prompt>
  </block>
  <noinput>
    <field name="confirm" type="boolean">
    <prompt> Do you get it? </prompt>
    <if cond="confirm=false"> <goto next="operator.vxml"> </if>
  </noinput>
 </form>
</vxml>
```

Figure 8: VoiceXML file for explaining transportation method.

```
<?xml version="1.0"?>
<vxml version="1.0">
 <menu>
   <prompt> We have <enumerate/> room.
            which room type do you prefer?</prompt>
   <choice next="#single"> single </choice>
   <choice next="#double"> double </choice>
   <choice next="#twin"> twin </choice>
 </menu>

 <form id="#single">
  <block> Room charge for single room is 10,000Yen. </block>
 </form>
 <form id="#double">
  <block> Room charge for double room is 15,000Yen. </block>
 </form>
 <form id="#twin">
  <block> Room charge for twin room is 15,000Yen. </block>
 </form>
</vxml>
```

Figure 9: VoiceXML file for explaining room type information.

[VoiceXMLForum 2000]
VoiceXMLForum. *Voice extensible markup language VoiceXML.* http://www.voicexml.org/, 2000.

[Araki et al. 1999] M. Araki, K. Komatani, T. Hirata, and S. Doshita. *A dialogue library for task-oriented spoken dialogue systems.* In *Proc. IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 1–7, 1999.