KEYWORDS AND WEIGHTING FOR PRODUCT SPECIFICATIONS EXTRACTION

KAZUTAKA SHIMADA, KOJI HAYASHI AND TSUTOMU ENDO

Department of Artificial Intelligence,
Kyushu Institute of Technology,
Kurume, Fukuoka 820-8502 Japan
{shimada, k_haya, endo}@philo.ai.kyutech.ac.jp

Product specifications contain many data. It is not, however, clear which ones are the characteristic data in them. We are developing a multi-specifications summarization system using extracted characteristic data from the product specifications. The specifications are written in a <TABLE> tag. The presence of the <TABLE> tag in an HTML document does not necessarily indicate the presence of specifications. Less than 30% of HTML <TABLE> tags are real tables in one particular domain. In this paper, we propose a method for keyword extraction for product specifications extraction. For PC and digital still camera specifications, we evaluate the performance for two keyword sets which are constructed by an entropy and a Bayes theorem based method.

Key words: Table Extraction, Keyword Extraction, Weighting

1. INTRODUCTION

As the World Wide Web rapidly grows, a huge number of online documents are easily accessible on the Web. Finding information relevant to user needs has become increasingly important. One of the useful online documents is specifications for equipment about products such as personal computers and digital still cameras. In general, their specifications are presented in tabular form as shown in Fig. 1. The specifications on the WWW are written in a <TABLE> tag. The presence of the <TABLE> tag in an HTML document does not necessarily indicate the presence of specifications. Less than 30% of HTML <TABLE> tags are real tables in one particular domain [1]. Since tables are an efficient way to express relational information, table extraction is a significant task for web mining, summarization and so on.

In this paper, we propose a method for keyword extraction for product specifications extraction. We evaluate the performance for two keyword sets, which are constructed by an entropy and a Bayes theorem based method. Figure 2 shows the process flow of the proposed table extraction. The process consists of the filtering and the extraction processes. The filtering is to extract Web pages including specifications. The extraction is to extract specifications from the filtered Web page. We evaluate our methods with PC and digital still camera specifications.

2. PRODUCT RANKING SYSTEM USING USER’S REQUESTS

Although specifications contain many kinds of data, it is not clear which ones are the characteristic data among them. For example, consider users who want to buy a personal computer. They retrieve product information that includes specifications from Web sites of many computer makers. However, it is difficult for users except some experts to select a suitable computer for their own purpose from the several specifications. The reasons are as follows:

© 2003 Pacific Association for Computational Linguistics
1. Each Web site provides its own product, and does not contain comparison with other maker’s products.
2. Web pages of each site have various styles, and it is not easy to compare them with other maker’s ones.
3. Extraction of characteristic-data and association of user’s requests with specifications of each product require technical knowledge.
To satisfy a user’s request, a Web-based system must integrate the information from the various sites into a single, coherent whole. Unfortunately, integrating information from diverse sources is very hard when information is presented in a simple structure [2].

The purpose of our study is to develop a multimedia summarization system. As the initial step, we focus on a table on the World Wide Web. We are developing a multi-specifications summarization system from multiple Web sites [5][6]. Figure 3 shows the process flow of our system. Figure 4 shows a snapshot of our system. Our system has 3 features: (1) scoring using 5 requests and attribute selection, (2) score re-calculation using relevance feedback, and (3) generation of a radar chart and
Japanese sentences from specifications. Since the system requires product specifications as input, specifications extraction is an important task for it.

3. RELATED WORK

There are several approaches to deal with HTML-based tables. Although Chen et al. have reported a method for mining tables from HTML documents, they employed heuristic rules for table extraction [1]. Constructing rules by handwork is costly. Wang et al. have reported a machine learning based approach for table extraction [8]. They evaluate two methods: decision tree learning and Support Vector Machines (SVM). The purpose of Chen et al. and Wang et al. is to extract tables from the Web. They do not, however, deal adequately with the usage of the extracted tables. We have verified the utility of the extracted table data using a multi-specifications summarization system [6].

On the other hand, Yoshida et al. have verified the utility of table data [9]. The purpose is to build ontologies from the World Wide Web via HTML tables. Our purpose is to extract the characteristic-data of each products by comparing several specifications with each other, and to present products suitable for a user's request.

4. KEYWORDS

We handle Web pages about computers and digital cameras as input. These pages are retrieved from multiple sites by a file-downloading software. Our system extracts keywords from them. Here we define keywords as follows:

1. Words in 1st column in a table;
2. Words which appear in a text of specific length;
3. Words which appear frequently in a document including specifications or not including specifications.

We handle the contents of the 1st <TD> in <TR> tags. If the contents consist of 25 characters or less, our system extracts it as keyword candidates. The condition is heuristic. We divide the keyword candidates into words by using the Japanese morphological analyzer ChaSen [3]. Weights of keywords fall into two categories: Keyword Weight (KW) and Noise-word Weight (NW). The KW is the weight of a keyword to extract tables and documents including specifications. The NW is the weight of a keyword to extract non-tables and documents not including specifications.

5. WEIGHTING

We employ two methods for weighting: entropy and Bayes theorem.

5.1. Entropy

First, we apply entropy to the weighting. Entropy is a measure of bias of term frequency [7]. We divide documents $D = (d_1, ..., d_N)$ into $D_{real}$ and $D_{no}$. $D_{real}$ denotes the documents including specifications, and $D_{no}$ denotes the documents not
including specifications. The weight of \( \text{term}_k \) in \( D_{\text{real}} \) and the weight of \( \text{term}_k \) in \( D_{\text{no}} \) are computed as:

\[
wr^\text{real}_k = \frac{w^D_{\text{real}}}{w^P_{\text{real}}}, \quad wr^\text{no}_k = \frac{w^D_{\text{no}}}{w^P_{\text{no}}},
\]

where

\[
w^D_{\text{real}} = \log \sum_{k=1}^{M} tf(t, k) + \sum_{i=1}^{M} \frac{tf(t, i)}{\sum_{j=1}^{M} tf(t, j)} \log \frac{tf(t, i)}{\sum_{j=1}^{M} tf(t, j)}.
\]

\( tf(t, i) \), \( tf(t, j) \) and \( tf(t, k) \) are the frequency of \( \text{term}_k \) in \( \text{document}_i \), \( \text{document}_j \) and \( \text{document}_k \) respectively. \( M \) is the number of documents in \( D_{\text{real}} \) or \( D_{\text{no}} \). The weight of \( \text{term}_k \) as KW is \( ws^\text{real}_k = df(t, D_{\text{real}}) \times wr^\text{real}_k \). The weight of \( \text{term}_k \) as NW is \( ws^\text{no}_k = df(t, D_{\text{no}}) \times wr^\text{no}_k \). \( df(t, D_{\text{real}}) \) and \( df(t, D_{\text{no}}) \) are the number of documents including \( \text{term}_k \) in \( D_{\text{real}} \) and \( D_{\text{no}} \) respectively. We employ the words of top ranks of \( ws^\text{real}_k \) and \( ws^\text{no}_k \) as the KWS and the NWs for the entropy method.

5.2. Bayes theorem

Next, we apply Bayes theorem to the weighting. The Bayes theorem is a probabilistic method [4]. The probability that \( \text{term}_k \) belongs to class \( C_i \) is given by:

\[
P(C_i|t) = \frac{P(C_i)p(t|C_i)}{p(t)},
\]

where \( C = \{ D_{\text{real}}, D_{\text{no}} \} \). We handle the probabilities as the weights of KWS and NWs. In other words, the \( ws^\text{real}_k \) is \( P(D_{\text{real}}|t) \) and the \( ws^\text{no}_k \) is \( P(D_{\text{no}}|t) \). The conditions of the KWSs are \( P(D_{\text{real}}|t) \geq 0.75 \) and \( df(t, D_{\text{real}}) \geq \frac{D^M_{\text{real}}}{2} \). The conditions of the NWs are \( P(D_{\text{no}}|t) \geq 0.75 \) and \( df(t, D_{\text{no}}) \geq \frac{D^M_{\text{no}}}{2} \). \( D^M_{\text{real}} \) and \( D^M_{\text{no}} \) are the number of documents in \( D_{\text{real}} \) and \( D_{\text{no}} \) respectively.

6. FILTERING

Filtering is to extract web pages including specifications. The filtering process is as follows:

1. Extract an area written in a \(<\text{TABLE}>\) tag from an HTML document \( d_i \).
2. Extract the contents \((\text{Cont})\) of \(<\text{TD}>\) tags in the \(<\text{TABLE}>\) tag.
3. Compute

\[
\text{Ratio}_{\text{real}} = \frac{\sum_{t \in \text{Cont}} ws^\text{real}_t}{\sum_{t \in \text{KW}} ws^\text{real}_t}, \quad \text{Ratio}_{\text{no}} = \frac{\sum_{t \in \text{Cont}} ws^\text{no}_t}{\sum_{t \in \text{NW}} ws^\text{no}_t}.
\]

4. Compute

\[
\text{Score}_{e_i} = \text{Ratio}_{\text{real}} \times \frac{\text{Ratio}_{\text{real}}}{\text{Ratio}_{\text{no}}}.
\]

5. If \( \text{Score}_{e_i} \) is more than or equal to a threshold \( th1 \), extract the HTML document \( d_i \).
Table 1. Dataset

<table>
<thead>
<tr>
<th>Product</th>
<th>Training</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>DocIncSpecs</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>DocNotIncSpecs</td>
<td>50</td>
</tr>
<tr>
<td>Digital Camera</td>
<td>DocIncSpecs</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>DocNotIncSpecs</td>
<td>50</td>
</tr>
</tbody>
</table>

Table 2. Keyword and Noise-word

<table>
<thead>
<tr>
<th>Product</th>
<th>Keyword</th>
<th>Noise-word</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>Entropy</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Bayes</td>
<td>19</td>
</tr>
<tr>
<td>Digital Camera</td>
<td>Entropy</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Bayes</td>
<td>28</td>
</tr>
</tbody>
</table>

7. EXTRACTION

Extraction is to extract specifications from the filtered web page. The extraction process is as follows:

1. Extract an area written in a <TABLE> tag.
2. Extract the contents (Cont) of <TD> tags in the <TABLE> tag.
3. If any KWs do not exist in the contents, search a next <TABLE> tag.
4. Compute $\text{Sum}_i = \sum_{i \in \text{Cont}} w_s^{i, \text{real}}$ for $\text{table}_i$.
5. Extract $\text{table}_i$ maximizing $\text{Sum}_i$.
6. If $\text{Sum}_i$ is more than or equal to a threshold $\text{th2}$, extract $\text{table}_i$ as specifications.

$$\text{th2} = \frac{\sum_{i \in \text{KW}} w_s^{i, \text{real}}}{2}.$$

8. EVALUATION

For PC and digital still camera specifications, we evaluated the performance of two keyword sets. The number of data is shown in Table 1. In Table 1, the DocIncSpecs and the DocNotIncSpecs denote documents including specifications and documents not including specifications, respectively. Both of them consist of specifications, other tables, texts and images. For digital cameras, the DocIncSpecs includes the specifications of video cameras and still cameras. The number of KWs and NWs is shown in Table 2. For the entropy method, we used the words of the top 30 and the top 15 as KWs and NWs. Figure 5 and 6 show examples of KWs and NWs for PCs and digital still cameras respectively.

Three performance measures Recall rate ($R$), Precision rate ($P$) and $F$-measure ($F$) are computed as follows:

$$\text{Recall (} R \text{)} = \frac{\# \text{ of extracted correct documents}}{\# \text{ of documents including specifications}}$$
surotto (slot) :: 272507.986
naizou (internal) :: 149048.428
gabu (external) :: 147305.761

kyasshu (cache) :: 1
saundo (sound) :: 1
surotto (slot) :: 0.97361

nichi (day) :: 13631.914
tsuki (month) :: 13631.912
gazou (image) :: 10515.892

kyasshu (cache) :: 1
saundo (sound) :: 1
surotto (slot) :: 0.97361

nichi (day) :: 13631.914
tsuki (month) :: 13631.912
gazou (image) :: 10515.892

douga (moving picture) :: 1
howaito (white) :: 1

KWs by entropy
NWs by entropy
KWs by Bayes
NWs by Bayes

Precision \( (P) = \frac{\text{# of extracted correct documents}}{\text{# of extracted documents}} \)

\( F - \text{measure} \ (F) = \frac{1}{\alpha \frac{P}{R} + (1 - \alpha) \frac{R}{P}} \)

For a filtering process, we set \( \alpha \) to 0.4 because we consider that \( (R) \) is more important than \( (P) \) in this process.

The experimental results for the filtering process are shown in Table 3. The Bayes theorem based method produced the best performance for both PCs and digital cameras. The \( F - \text{measure} \) of the entropy method was lower than that of the Bayes theorem based method because the weight of the entropy method was not normalized. In other words, the range of the weights by the Bayes theorem based method is from 0 to 1 because the method is a probabilistic model. We expanded the weights of the Bayes theorem based method using the following formulas:

\[ w_{st}^{real} = \frac{P(D_{real}|t)}{P(D_{no}|t)}, \quad w_{st}^{no} = \frac{P(D_{no}|t)}{P(D_{real}|t)} \]

The results for PCs by this formula are \( (R) = 99.0\%, \ (P) = 98.0\%, \) and \( (F) = 98.6\% \).
Table 3. Filtering

<table>
<thead>
<tr>
<th>Product</th>
<th>Method</th>
<th>th1</th>
<th>Recall</th>
<th>Precision</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>Entropy</td>
<td>0.25</td>
<td>100.0%</td>
<td>81.3%</td>
<td>91.6%</td>
</tr>
<tr>
<td></td>
<td>Bayes</td>
<td>0.50</td>
<td>100.0%</td>
<td>99.0%</td>
<td>99.6%</td>
</tr>
<tr>
<td>Digital Camera</td>
<td>Entropy</td>
<td>0.80</td>
<td>94.6%</td>
<td>75.9%</td>
<td>86.1%</td>
</tr>
<tr>
<td></td>
<td>Bayes</td>
<td>0.55</td>
<td>96.8%</td>
<td>94.7%</td>
<td>96.0%</td>
</tr>
</tbody>
</table>

Table 4. Extraction

<table>
<thead>
<tr>
<th>Product</th>
<th>Method</th>
<th>th1</th>
<th>Recall</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>Entropy</td>
<td>0.25</td>
<td>93.0%</td>
<td>96.9%</td>
</tr>
<tr>
<td></td>
<td>Bayes</td>
<td>0.50</td>
<td>95.0%</td>
<td>100.0%</td>
</tr>
<tr>
<td>Digital Camera</td>
<td>Entropy</td>
<td>0.80</td>
<td>80.7%</td>
<td>86.6%</td>
</tr>
<tr>
<td></td>
<td>Bayes</td>
<td>0.55</td>
<td>82.8%</td>
<td>95.1%</td>
</tr>
</tbody>
</table>

These results show the significance of normalization.

The experimental results for the extraction process are shown in Table 4. The Bayes theorem based method also produced the best performance. Although DocNotIncSpecs included vague specifications such as video cameras, we obtained high recall and precision rates by the proposed method.

9. CONCLUSIONS

Table extraction in web documents is an interesting problem with many applications. We extracted product specifications as input for a multi-specifications summarization system. We evaluated two keyword sets for table extraction algorithm. We obtained high recall and precision rates, especially PC specifications. Our future work includes handling more product specifications and evaluating other weighting algorithm.

REFERENCES


