Product Specifications Summarization and Product Ranking System using User’s Requests

Kazutaka SHIMADA and Tsutomu ENDO

Department of Artificial Intelligence, Kyushu Institute of Technology, Iizuka, Fukuoka 820-8502 Japan

Abstract. This paper proposes a method to integrate computer specifications retrieved from multiple Web sites, to extract characteristic-data of each computer based on integrated information, and to present products suitable for a user’s request. The specifications written in HTML are converted into normal forms called table structure. The quantitative attributes such as speed, capacity and dimensions are extracted by comparing them with the mean or mode of all sample data, and the qualitative ones such as kind of processor and graphics chip are extracted using knowledge provided manually. The recommended products are dynamically determined from the extracted data by a user’s request and relevance feedback. Moreover, a radar chart and Japanese sentences are generated from specifications. Experimental results show the effectiveness of our method.

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. Although they 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:

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[1].

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, focusing on personal computer products [2]. This paper proposes a method to integrate specifications presented in tabular form, to extract characteristic-data of each computer based on integrated information, and to present computers for a user’s request. Figure 2 shows the process flow of our system. First, Web pages are retrieved from multiple sites by a file-downloading software. Product specifications are extracted from the Web pages. Then HTML-based specifications are converted into normal forms called table structure. These processes are available for other product specifications. Next, characteristic-data are extracted from the table structures. Finally, the recommended computers are dynamically determined by scoring these data according to a user’s request and relevance feedback. Moreover, a radar chart and Japanese sentences are generated from specifications. We provide domain specific knowledge manually for the characteristic-data extraction and the sentence generation. The knowledge is the following: (1) Normal forms of attribute names, (2) Correspondence of an attribute with a unit, (3) The relevant technical terms such as “PCI” and “USB”, (4) Kind of processor (e.g., Pentium III and Celeron), and (5) Relations between attributes and a user’s request, (6) Frames for sentence generation.
2 Table Detection

Here we handle Web pages about computers as input. These pages are retrieved from multiple sites by a file-downloading software. The contexts of the retrieved pages are not only tables but also text and images. The \(<Table>\) tag in an HTML document is not always a real table because it is often employed for a layout of the Web page. In order to extract the specifications from retrieved pages, we extract keywords and calculate their weights. We apply entropy to the weight. 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 \(term_t\) is computed as:

\[
ws_t = \frac{w_t^{D_{real}}}{w_t^{D_{no}}}
\]

where

\[
w_t^{D_{no}} = \log \sum_{k=1}^{M} tf(t, k) + \sum_{i=1}^{M} \frac{tf(t, i)}{\sum_{j=1}^{M} tf(t, j)} \log \frac{tf(t, i)}{\sum_{j=1}^{M} tf(t, j)}
\]

\(tf(t, i), tf(t, j)\) and \(tf(t, k)\) are the frequency of \(term_t\) in \(document_i\), \(document_j\) and \(document_k\) respectively. \(M\) is the number of documents in \(D_{real}\) or \(D_{no}\).

First, our system extracts documents including specifications from downloaded Web pages. If \(\sum_{t \in \Omega} ws_t\) is more than or equal to a threshold, the system extracts the document. The threshold is \(\frac{\sum ws_t}{2}\). Next, specifications are extracted from the document. The system computes \(Score_j = \sum_{t \in table_j} ws_t \times num_j\) for \(table_j\) in the document and extracts the \(table_j\) maximizing \(Score_j\). The \(num_j\) is the number of keywords in \(table_j\). If \(Score_j\) is more than or equal to a threshold, the system extracts the \(table_j\) as specifications. The threshold is \(\frac{\sum ws_t}{2} \times \frac{\# of keywords}{2}\). See [25] for the other methods and evaluation of table detection.

3 Table Structure Conversion

Specifications are expressed in the form of a two-dimensional table. Generally, the first column corresponds to the attribute of a PC. The rest of columns correspond to the data about each PC, and the cell in \(i\)-th row shows the value of the \(i\)-th attribute.

The serious problem in these tables is that the style of description in each cell is not standardized as follows: (1) the kind and name of attributes are not standardized, (2) some cell contains two or more values, (3) some attribute has subcategories (e.g., “Memory” in Fig. 3 (a)), and (4) two or more cells which contain the same value are unified (e.g., “256MB” in Fig. 3 (a)).

3.1 Definition of Table Structure

To solve above problems, we define a normal form called table structure. The table structure is a set of simple ternary lists:

\[
\text{(Nam Atr Val)}
\]

where \(Nam\), \(Atr\), and \(Val\) are a model name, an attribute name, and a value respectively. As regards most specifications of products, \(Nam\) is located at the top of ones and \(Atr\) is located at the left side of ones. \(Nam\) and \(Atr\) are often represented by a list form.
3.2 Algorithm for Conversion

An algorithm to convert HTML-based specifications as shown in Fig. 3 (b) into table structures is as follows:

1. Decompose a unified cell by HTML tags, ROWSPAN and COLSPAN. Figure 4 shows an example of reformulated Fig. 3 (a).

2. Let $c(i, j)$ denote a cell in the $i$-th row and $j$-th column. $(\text{Nam}_k, \text{Atr}_k, \text{Val}_k)$ denotes the $k$-th list in table structures. Set $k = 1$. For each $j$ ($1 < j$), do the following substeps:
   2.1 Set $\text{Nam}_k = c(1, j)$ (i.e., the model name of a PC is set to Nam).
   2.2 For each $i$ ($1 < i$),
      (1) Set $\text{Atr}_k = c(i, 1)$ (i.e., the $i$-th attribute name is set to Atr).
      (2) Set $\text{Val}_k = c(i, j)$ (i.e., the value of the $i$-th attribute is set to Val).
      (3) Set $k = k + 1$.

3. For each list, do the following substeps using appropriate knowledge:
   3.1 Transform a word in a list into the normal form. We employ a manually constructed dictionary for the transformation. The number of keywords in
the dictionary is 50.

Ex. Monitor, Screen ⇒ Display

3.2 If the element Atr contains numerals with a unit such as “1024×768 dpi”, transfer them to the element Val.

Ex. (PC1 (Resolution 1024×768dpi) ○) ⇒
(PC1 Resolution 1024×768dpi)

3.3 If the element Val contains two or more values, divide the list into several lists using symbols such as “,” and “;”.

Ex. (PC1 Interface (USB×2, IEEE×1)) ⇒
(PC1 Interface USB×2)
(PC1 Interface IEEE×1)

3.4 If the element Val contains symbols with numerals and keywords such as “USB ×2”, transform the Atr and the Val.

Ex. (PC1 Interface USB×2) ⇒
(PC1 (Interface USB) 2)

3.5 Parse the particular notations such as “[ ]” and “…” and rewrite them into normal forms.

Ex. (PC1 (Bays total [free]) 5[2]) ⇒
(PC1 (Bays total) 5)
(PC1 (Bays free) 2)

Figure 5 shows an example of table structure converted from HTML data (Fig. 3 (b)).

4 Characteristic-data Extraction & Summarization

Specifications contain many attributes and values about PCs. It is not, however, clear which ones are the characteristic-data. Our system extracts the attributes and values that characterize each PC. The attribute is classified into two categories: quantitative and qualitative. The typical example is listed in Table 1. Figure 6 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.

4.1 Characteristic-data Extraction using Quantitative Attributes

For quantitative attributes, the characteristic-data are extracted by comparing each value. The attributes that have the same value in all PCs are rejected. Table 2 shows
the classification of unit for the comparison. There are two comparison processes: (1) extraction of attributes with the maximum or minimum value and (2) comparison with a standard value. The standard value is the mean or mode computed from sample data. Table 3 shows the classification of unit for computing the standard value. The following preprocessing is performed before computing the mean or mode:

- The unit of value is standardized: (mm, cm), (MHz, GHz), and (KB, MB, GB).
- As for the data with the range, the maximum or minimum value is employed:
  (PC1 (Dimensions Height) 38-40mm) ⇒
  (PC1 (Dimensions Height) 38mm).

Each value obtains a score by comparing it with standard value. We define the scores: minimum, standard, and maximum points are 0, 5, and 10 points respectively. Our system calculates the value per 1 point from them. The calculation is exemplified in Fig. 7. Assume that “500MHz”, “600MHz”, and “1.1GHz” are the minimum, standard,
Table 3: The classification of units for standard value.

<table>
<thead>
<tr>
<th>Mean</th>
<th>W, yen, $, mm, hours, Kg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode</td>
<td>MHz, KB, MB, GB, colors, inch</td>
</tr>
</tbody>
</table>

1.1GHz (Maximum Value): 10pts.

**Ex.** The CPU clock of a PC is 800MHz
The score is 7pts.

100MHz per 1pt.

600MHz (Standard Value): 5pts.

20MHz per 1pt.

500MHz (Minimum Value): 0pts.

Figure 7: The calculation of the score.

and maximum value, which were calculated from all PCs, respectively. If the clock speed of a PC is “800MHz”, the PC obtains 7 points.

Most data in specifications is a numerical value. By changing Table 2 and Table 3, our system can, in principle, extract the characteristic-data from the specifications of other products such as digital still cameras and cellular phones.

4.2 Characteristic-data Extraction using Qualitative Attributes

For qualitative attributes, we employ domain knowledge, which consists of keywords such as processor names, for the characteristic-data extraction. The number of keywords in domain knowledge is 23. The characteristic-data are extracted as follows:

1. Search a keyword from all table structures with qualitative attributes.
2. If the keyword exists in all products, it is not extracted as the characteristic-data.
3. If the keywords possess weight, score the weight to the PC with the keywords (e.g., Pentium4: 4, Pentium3: 3, and Celeron: 2).
4. Extract the data exceeding a threshold value as the characteristic-data.

4.3 Score Calculation by a User’s Request

The characteristic-data extraction process is static. To find PCs that relate to a user’s request, we define relationships between a user’s request and attributes. Table 4 shows examples of the relationships. Our system can handle their 5 requests. Each attribute possesses weight. Moreover, a user can settle the weight of each attribute (from -1 to 4). Figure 8 shows the window for attribute selection. There are relationships between attributes. For example, “Weight” is related to “Dimensions” and “Battery Life”. Their relationships are defined manually. Table 5 shows examples of the relationships between attributes. By clicking the “Related Items” button in Fig.8,
Table 4: The relationships between a user’s request and attributes

<table>
<thead>
<tr>
<th>Request</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>High performance</td>
<td>CPU, Memory, Display, HDD, Interface</td>
</tr>
<tr>
<td>High graphics performance</td>
<td>Display, Graphics, CPU, Memory</td>
</tr>
<tr>
<td>Mobile PC</td>
<td>Battery life, Dimensions, Weight</td>
</tr>
<tr>
<td>Practical use</td>
<td>CPU, HDD, Price, Interface, Software</td>
</tr>
<tr>
<td>Budget PC</td>
<td>Price, Software, Memory, CPU</td>
</tr>
</tbody>
</table>

![Figure 8: The window for attribute selection.](image)

Our system extends automatically the weights of the related attributes using the relationships, based on attributes selected by the user. The weight of a related attribute is calculated as follows:

\[
uw(j) = \begin{cases} 
0.5 & (uw(i) = 1) \\
-0.5 & (uw(i) = -1) \\
uw(i) - 1 & \text{ (The others)} \end{cases}
\]

where \(i\) and \(j\) are the attribute selected by a user and the related attribute respectively. \(uw(j)\) is the weight of \(j\) by user’s selections. Assuming that a user sets the weight of an attribute “Weight” to 2 and clicks the “Related Items” button, our system sets the ones of “Dimensions” and of “Battery Life” to 1.

The score calculation process is as follows:

1. Select the table structures with attributes relating to a user’s request.
2. For each selected PC, compute

\[
score(c, r) = \frac{\sum_{k=1}^{n}(w(a_k, r) + uw(a_k)) \times pt(a_k, c)}{\sum_{k=1}^{n} w(a_k, r)}
\]

where \(c\), \(r\) and \(a_k\) are a PC, a user’s request and an attribute respectively. \(w(a_k, r)\) is the weight of \(a_k\) in the request. We define \(w(a_k, r)\) manually. \(pt(a_k, c)\) is the score calculated in Sect. 4.1 and 4.2.

3. Extract the PCs exceeding a threshold value.
4. Return them as the recommended computers in descending order for the score.
Table 5: Examples of the relationships between attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Related attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Memory (Std. and Max.), HDD</td>
</tr>
<tr>
<td>Display</td>
<td>VRAM, Resolution, Colors</td>
</tr>
<tr>
<td>USB</td>
<td>IEEE, PC card</td>
</tr>
<tr>
<td>Weight</td>
<td>Dimensions, Battery life</td>
</tr>
</tbody>
</table>

![Radar chart](image)

Figure 9: A radar chart.

### 4.4 Relevance Feedback

A single request is often insufficient because the weight is static. On the other hand, a user is often aware of the product which he/she needs by browsing the result of the search. Relevance feedback is an iterative process to improve the retrieval effectiveness [3]. Our system updates each weight during the relevance feedback. Assuming that a query consists of the weights of each attribute, the initial query is $Q_0 = (w(a_1, r), ..., w(a_n, r))$. A new query $Q_1$ is computed as:

$$Q_1 = Q_0 + \alpha \sum_{i=1}^{N^+} D^+_i - \beta \sum_{i=1}^{N^-} D^-_i$$

where $D^+$ and $D^-$ are the vector for the relevant and non-relevant PC respectively. $N^+$ and $N^-$ are the number of relevant and non-relevant PC chosen respectively. $\alpha$ and $\beta$ tune the importance of relevant and non-relevant attributes respectively.

### 4.5 Generation of a radar chart and Japanese sentences from specifications

Our system can generate a radar chart and Japanese sentences from the characteristic-data of selected products by a user. The radar chart is generated by scores calculated in previous subsections. Figure 9 shows an example of a generated radar chart. “CPU”, “Memory”, “HDD”, “Device”, “Display”, “Battery”, “Weight” and “Price” are selected as default axes for the radar chart. A user can select each axis from attributes in specifications.
Table 6: Relationships between the topics and the attributes.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance</td>
<td>CPU, Memory, Hard disk, etc.</td>
</tr>
<tr>
<td>Scalability</td>
<td>PCI, USB, PC card, etc.</td>
</tr>
<tr>
<td>Image processing</td>
<td>Graphics Chipset, Image processing soft, etc.</td>
</tr>
<tr>
<td>Display</td>
<td>Screen size, Resolution, VRAM, etc.</td>
</tr>
<tr>
<td>User-friendliness</td>
<td>Key size, Input device, etc.</td>
</tr>
<tr>
<td>Mobility</td>
<td>Weight, Dimensions, Battery life, etc.</td>
</tr>
<tr>
<td>Communication</td>
<td>Modem type, LAN, etc.</td>
</tr>
<tr>
<td>Sound</td>
<td>Speaker, Sound board, etc.</td>
</tr>
<tr>
<td>Soft</td>
<td>OS, Bundled software, etc.</td>
</tr>
</tbody>
</table>

Figure 10: Document structure.

We define sentence generation frames (SG) and explanation sentences (ES) for sentence generation. Figure 10 shows document structure in our system. The number of topics is 9. Table 6 shows relationships between the topics and the attributes in specifications. Examples of SGs are as follows:

- **Topic**:
  - [Topic] no {sugureta or you} [Nam]. (Japanese)
  - [Nam] is excellent in [Topic]. (English)

- **Text**:
  - [Nam] wa [Atr] ni [Val] wo {tousai or saiyou}. (Japanese)
  - [Nam] is equipped with [Atr] of [Val]. (English)

[Topic], [Nam], [Atr], and [Val] are slots. Sentences are generated from SG of which the slots are filled with characteristic-data. The number of SGs is 27 frames. ESs are employed to generate additional information to supplement for the generated sentence by SGs. ESs possess the condition for generation, but do not possess any slots. An example of ESs is as follows:

- **Condition**: [Val] = “USB”

- **ES**: USB ha syuenkiki wo tunagu interface desu. (Japanese)
  - Universal Serial Bus, or USB, is an interface for connecting peripherals to your PC. (English)

The number of ESs is 35 sentences.
Our system re-calculates the scores of selected products by a user using the method described in subsection 4.1 and 4.2. [Nam] of SG for a topic is filled with the name of product of which the total of the characteristic-data of the attributes belonging to the topic is the maximum. If the score of characteristic-data is 5 or more, our system generates an additional sentence such as “[Nam] ha hikakuteki yoi (|[Nam] is comparatively good).” Figure 11 shows an example of sentence generation processing.

5 Evaluation

In this section, we present experimental results to evaluate our system, and compare our approach with related work.

5.1 Table Detection and Conversion

To evaluate the table detection process, we used 200 documents from 5 Web sites. The number of documents including specifications was 100. The number of documents not including specifications was 100. We used 100 documents, which are 50 documents including specifications and 50 documents not including specifications, for weighting. We used recall and precision rates for the evaluation. The recall rate (R) and the precision rate (P) are computed as:

\[
R = \frac{\text{The Number of Documents Extracted Correctly}}{\text{Total Number of Correct Documents}}
\]

\[
P = \frac{\text{The Number of Documents Extracted Correctly}}{\text{Total Number of Extracted Documents}}
\]

The recall rate was 97% and the precision rate was 95%. We obtain the high recall rate and the high precision rate.

To evaluate the table conversion process, we used the 100 specifications including 247 products. The accuracy for the table conversion was 94%. The reason for the failure of the conversion is an error in HTML grammar: omission of </TD> tags or </TR> tags in a <TABLE> tag. If the HTML of specifications is described correctly, the accuracy for the table conversion rises to 100%.
<table>
<thead>
<tr>
<th></th>
<th>PC1</th>
<th>PC2</th>
<th>PC3</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>800MHz</td>
<td>700MHz</td>
<td>700MHz</td>
</tr>
<tr>
<td>Memory</td>
<td>Std. 64MB</td>
<td>64MB</td>
<td>64MB</td>
</tr>
<tr>
<td></td>
<td>Max. 128MB</td>
<td>128MB</td>
<td>256MB</td>
</tr>
<tr>
<td>Hard Drive</td>
<td>external</td>
<td>5GB</td>
<td>10GB</td>
</tr>
<tr>
<td>Optical Device</td>
<td>external</td>
<td>internal</td>
<td></td>
</tr>
<tr>
<td>Display</td>
<td>11.3” TFT</td>
<td>12.1” TFT</td>
<td>13.3” TFT</td>
</tr>
<tr>
<td>VRAM</td>
<td>2.5MB</td>
<td>2MB</td>
<td>2MB</td>
</tr>
<tr>
<td>Resolution</td>
<td>1024 × 768 pixels</td>
<td>800 × 600 pixels</td>
<td>800 × 600 pixels</td>
</tr>
<tr>
<td>PC card</td>
<td>TYPE II × 1</td>
<td>TYPE II × 1</td>
<td>TYPE II × 1</td>
</tr>
<tr>
<td>Interface</td>
<td>USB × 1, IrDA × 1, LAN × 1</td>
<td>USB × 1, IrDA × 1</td>
<td>USB × 2</td>
</tr>
<tr>
<td>Key size</td>
<td>17mm</td>
<td>18mm</td>
<td>17mm</td>
</tr>
<tr>
<td>Battery Life</td>
<td>Std. 1.8 hrs</td>
<td>2.5 hrs</td>
<td>1.5 hrs</td>
</tr>
<tr>
<td></td>
<td>Max. 4 hrs</td>
<td>5 hrs</td>
<td>7 hrs</td>
</tr>
<tr>
<td>Dimensions</td>
<td>270mm × 351mm × 29mm</td>
<td>270mm × 224mm × 23mm</td>
<td>257mm × 223mm × 29mm</td>
</tr>
<tr>
<td>Weight</td>
<td>1.6kg</td>
<td>1.56kg</td>
<td>1.5kg</td>
</tr>
</tbody>
</table>

Figure 12: Specifications in the experiment

5.2 Characteristic-data Extraction

We carried out the experiment of the characteristic-data extraction using the table structures converted from 66 specifications. Figure 12 shows an example, where each PC is the product of a different maker. The extracted values as the characteristic-data are shown in Table 7. In order to verify the accuracy for them, we compared them with a review in a magazine about PCs. The characteristics of the PCs in Fig. 12 are reviewed as follows:

PC1 Basic specs and a display are high performance. The user-friendliness and the mobility are comparatively good.

PC2 The user-friendliness.

PC3 The mobility and the scalability.

The clock speed of a processor and the capacity of a hard disk are the measures about basic specs. “Resolution” and the capacity of “VRAM” are the measures about performance of a display. Extracting the value of “Dimensions”, “Weight” and “Battery Life” is appropriate because they are related to mobility. The maximum capacity of memory and expansion options such as “USB” are the measures about scalability. “Key size” is related to user-friendliness.

The number of characteristic-data extracted from 66 specifications is 365. Table 8 shows the correctness of the characteristic-data. Eval(1) in Table 8 denotes the number of characteristic-data, which we judged to be correct by the review. Eval(2) in Table 8 denotes the number of characteristic-data, which we could not judge to be correct or incorrect by the review (e.g., “IrDA” in Table 7). Although we could not judge the correctness of the characteristic-data in Eval(2) by the review, we considered that they were appropriate.

5.3 User’s Request and Relevance Feedback

We evaluated a prototype system using a user’s request and relevance feedback. We used 38 specifications about notebook PCs. Our system can handle 5 requests: (1) High Performance, (2) High Graphics Performance, (3) Mobile PC, (4) Practical Use, and (5)
Table 7: Extracted characteristic-data

<table>
<thead>
<tr>
<th>PC Name</th>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC1</td>
<td>CPU</td>
<td>800MHz</td>
</tr>
<tr>
<td></td>
<td>HDD</td>
<td>15GB</td>
</tr>
<tr>
<td></td>
<td>Resolution</td>
<td>1024×768 pixels</td>
</tr>
<tr>
<td></td>
<td>VRAM</td>
<td>2.5MB</td>
</tr>
<tr>
<td></td>
<td>Interface:IrDA</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Interface:LAN</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Dimensions</td>
<td>270mm × 215mm × 29mm</td>
</tr>
<tr>
<td></td>
<td>Weight</td>
<td>1.6kg</td>
</tr>
<tr>
<td></td>
<td>Battery Life Max.</td>
<td>11hrs</td>
</tr>
<tr>
<td>PC2</td>
<td>HDD</td>
<td>15GB</td>
</tr>
<tr>
<td></td>
<td>Display</td>
<td>12.1” TFT</td>
</tr>
<tr>
<td></td>
<td>Interface:IrDA</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Key size</td>
<td>18mm</td>
</tr>
<tr>
<td></td>
<td>Battery Std.</td>
<td>2.5hrs</td>
</tr>
<tr>
<td>PC3</td>
<td>Memory Max.</td>
<td>256MB</td>
</tr>
<tr>
<td></td>
<td>Optical Device</td>
<td>Internal</td>
</tr>
<tr>
<td></td>
<td>Interface:USB</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Dimensions</td>
<td>257mm × 223mm × 29mm</td>
</tr>
<tr>
<td></td>
<td>Weight</td>
<td>1.5kg</td>
</tr>
</tbody>
</table>

Table 8: Correctness of characteristic-data

<table>
<thead>
<tr>
<th></th>
<th>Correctness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eval(1)</td>
<td>319/365</td>
</tr>
<tr>
<td>Eval(2)</td>
<td>46/365</td>
</tr>
</tbody>
</table>

Budget PC. We compared the result with recommended PCs in the magazine “Nikkei Best PC” [4]. Table 9 shows the ranking in the magazine of the product selected as the 1st by our system and the ranking in our system of the product selected as the 1st by the magazine. For the request (3) and (5), other results by our system also corresponded with the products ranked highly in the magazine. For the request (1) and (4), we obtained sufficient accuracy. Although some products ranked highly by our system did not correspond with the ones in the magazine for the request (2), two PCs recommended in the magazine were included in top five products of our result.

Next, we applied relevance feedback to the results. α and β for the relevance feedback were $\frac{1}{N^+}$ and $\frac{1}{N^-}$ respectively. The formula is well-known as Rocchio’s formula [5]. Figure 13 shows an example of the recall-precision rate of the results using relevance feedback. In the case of Fig. 13, a user chose “Mobile PC” as a user’s request first.

Table 9: The ranking in the magazine and our system.

<table>
<thead>
<tr>
<th>Request</th>
<th>The ranking in the magazine</th>
<th>The ranking in our system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request (1)</td>
<td>4th</td>
<td>3rd</td>
</tr>
<tr>
<td>Request (2)</td>
<td>8th</td>
<td>5th</td>
</tr>
<tr>
<td>Request (3)</td>
<td>1st</td>
<td>1st</td>
</tr>
<tr>
<td>Request (4)</td>
<td>4th</td>
<td>4th</td>
</tr>
<tr>
<td>Request (5)</td>
<td>1st</td>
<td>1st</td>
</tr>
</tbody>
</table>
Here, assume that the user attached importance to portability: the user put weight on the value of “Dimensions”, “Weight” and so on rather than “CPU”, “Memory” and so on. Next, the user judged the top 5 products based on the intention: relevant or non-relevant. The system re-calculated the score using $Q_1$ obtained by user’s feedback. “$Q_0$ with user’s weights” in Fig. 13 denotes the score calculated using attribute selection (Sect.4.3). The user set the weight of an attribute “Weight” to 3, and clicked the “Related Items” button. As a result, the recall-precision rate improved dramatically. The experimental results show the effectiveness of our system.

5.4 Generated sentences

We evaluated generated sentences with 8 graduate students. Figure 14 shows an example of the generated sentences. We employed 3 generated documents for the evaluation. The evaluation criteria of the generated sentences were as follows:

Eval(1) The grammatical accuracy.
Eval(2) The textual coherence.
Eval(3) The redundancy of expression.
Eval(4) The legibility.

8 graduate students classified the generated sentences into the following 5 categories:

Bad : 1pts.
Below average : 2pts.
Fair : 3pts.
Good : 4pts.
Excellent : 5pts.

Table 10 shows experimental results. Generated sentences were sufficient for summarization.
5.5 Related Work

Most of information extraction systems extract data from natural language text such as a newspaper article [6]-[10]. In their text-based systems, a large number of templates for pattern matching are necessary. One approach is to extract data from several articles [11]. The number of templates for the extraction system increases further. Moreover, newspaper articles lack information because they are a kind of summarized data. Therefore, specifications are among the best sources of data about products. As regards the need to change dictionaries in the extraction process, our system is easier than text-based systems because we employ units such as “MHz” and “GB” to extract characteristic-data from specifications.

On the other hand, there are several approaches to extract information using document structure such as itemization and tabular forms. Traditionally, they handle a document image [12] or plain text. Ng et al. have reported an approach that learns to recognize tables in free text [13]. Sato et al. have proposed a method for automatic generation of digests from the NetNews [14]. Kawai et al. have proposed a method for automatic extraction of relational information from itemized text [15]. However, they are different from table forms that we handle.

There are several approaches to deal with HTML-based documents. Although Chen et al. have reported a method for mining tables from HTML documents, their systems analyze only one table [16]. Hammer et al. have proposed a grammar-based tool for converting HTML pages into database objects [17]. They do not, however, deal adequately with the usage of structured data from tables. There is an approach to integrate several tables [18]. 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.
As regards summarization, most of them deal with texts [19]. Our system summarizes tables into sentences. There are many report generation systems such as stock market summarization and textual weather forecasts from databases [20]. Our system performs not only text generation, but also radar chart generation and presentation of the product rankings.

There are many shopbots on the World Wide Web [21], [22]. Most of them compare only the prices of products with each other. Chai [23] and Budzikowska [24] have proposed a conversational dialog system for online shopping. Their system, however, compares products with same maker’s ones. Our system can present the recommended products from several maker’s specifications using a user’s request and relevance feedback. The advantages of our system are as follows:

- A table-based information extraction system can be built with fewer patterns than a text-based system.
- It deals with specifications that contain many kinds of information about products.
- It can integrate several specifications retrieved from multiple sites by converting the HTML-based ones into normal forms called table structure.
- It can extract characteristic-data of each product, and find the products relevant to a user’s request from multiple specifications.
- It can generate a radar chart and sentences as a summary from specifications.

6 Conclusions

In this paper, we have proposed a method of information extraction from specifications on the Web. Our system presents the recommended products using a user’s request and relevance feedback. Moreover, a radar chart and Japanese sentences are generated from specifications. We obtained high recall and precision rates for the table detection process. The accuracy of the table conversion of HTML-based specifications into table structure was 94%. Most of the extracted characteristic-data were appropriate. We verified the effectiveness of our system. Our proposed system can be applied to specifications of other domain such as digital still cameras and cellular phones [26].

Some web sites and agents that compare some products already exist on the Web. However, most of them deal with only the prices of products. In comparison with traditional text-based information extraction systems, our system obtains much information because specifications contain more information than an article. Future work will include (1) construction of domain knowledge by machine learning, (2) information retrieval through man-machine dialogue, and (3) integration with other sources such as product images.

References


