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.

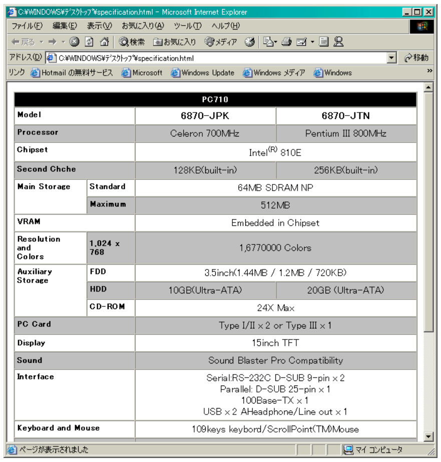
1 Introduction

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


C:¥WINDOWS¥ﾃﾞｽｸﾄｯﾌﾟ¥specification.html - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) お気に入り(A) ツール(T) ヘルプ(H)
戻る 検索 お気に入り メディア
アドレス(D) C:¥WINDOWS¥ﾃﾞｽｸﾄｯﾌﾟ¥specification.html 移動
リンク Hotmail の無料サービス Microsoft Windows Update Windows メディア Windows
ページが表示されました
マイ コンピュータ

PC710			
Model		6870-JPK	6870-JTN
Processor		Celeron 700MHz	Pentium III 800MHz
Chipset		Intel$^{(R)}$ 810E	
Second Chche		128KB(built-in)	256KB(built-in)
Main Storage	Standard	64MB SDRAM NP	
	Maximum	512MB	
VRAM		Embedded in Chipset	
Resolution and Colors	1,024 x 768	1,6770000 Colors	
Auxiliary Storage	FDD	3.5inch(1.44MB / 1.2MB / 720KB)	
	HDD	10GB(Ultra-ATA)	20GB (Ultra-ATA)
	CD-ROM	24X Max	
PC Card		Type I/II × 2 or Type III × 1	
Display		15inch TFT	
Sound		Sound Blaster Pro Compatibility	
Interface		Serial:RS-232C D-SUB 9-pin × 2 Parallel: D-SUB 25-pin × 1 100Base-TX × 1 USB × 2 AHeadphone/Line out × 1	
Keyboard and Mouse		109keys keybord/ScrollPoint(TM)Mouse	

Figure 1: Specifications of products.

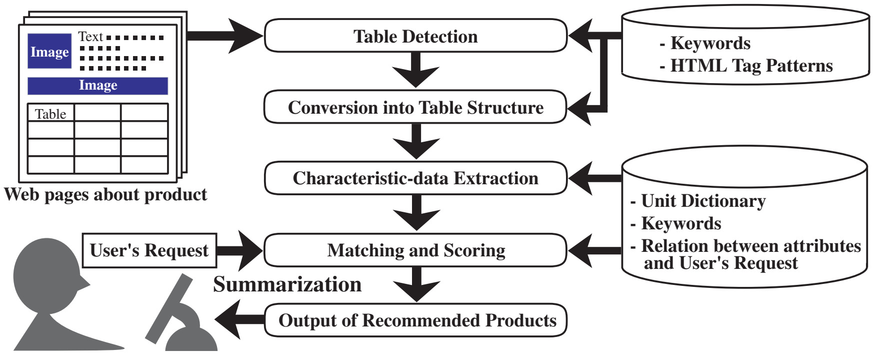


Figure 2: Outline of our system.

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 contents 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

$$w_t^{D_{type}} = \log \sum_{k=1}^{M} tf(t,k) + \sum_{i=1}^{M} \frac{tf(t,i)}{\sum_{j=1}^{M} tf(t,j)} \log \frac{tf(t,i)}{\sum_{j=1}^{M} tf(t,j)}$$

$tf(t,i)$, $tf(t,j)$ 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 $\sum_{t \in d_i} ws_t$ 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:

(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.

Model Name		PC1	PC2
CPU		400MHz	450MHz
Memory	Std	64MB	128MB
	Max	256MB	
	VRAM	4MB	

(a)

```
<table border="1">
   <tr>
       <td colspan="2"> Model Name</td>
       <td>PC1</td>
       <td>PC2</td>
   </tr>
   <tr>
       <td colspan="2">CPU</td>
       <td>400MHz</td>
       <td>450MHz</td>
   </tr>
   <tr>
       <td rowspan="3">Memory</td>
       <td>Std</td>
       <td>64MB</td>
       <td>128MB</td>
   </tr>
   <tr>
       <td>Max</td>
       <td colspan="2">256MB</td>
   </tr>
   <tr>
      <td>VRAM</td>
      <td colspan="2">4MB</td>
   </tr>
</table>
```

(b)

Figure 3: Specifications written in HTML.

Model Name		PC1	PC2
CPU		400MHz	450MHz
Memory	Std	64MB	128MB
Memory	Max	256MB	256MB
Memory	VRAM	4MB	4MB

Figure 4: Decomposed specifications.

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. (Nam$_k$, Atr$_k$, 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 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 Atr$_k$ = c(i, 1) (i.e., the i-th attribute name is set to Atr).
      (2) Set 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

```
(PC1  CPU 400MHz)
(PC1 (Memory Std) 64MB)
(PC1 (Memory Max) 256MB)
(PC1 (Memory VRAM) 4MB)
(PC2  CPU 450MHz)
(PC2 (Memory Std) 128MB)
(PC2 (Memory Max) 256MB)
(PC2 (Memory VRAM) 4MB)
```

Figure 5: Table structures.

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)

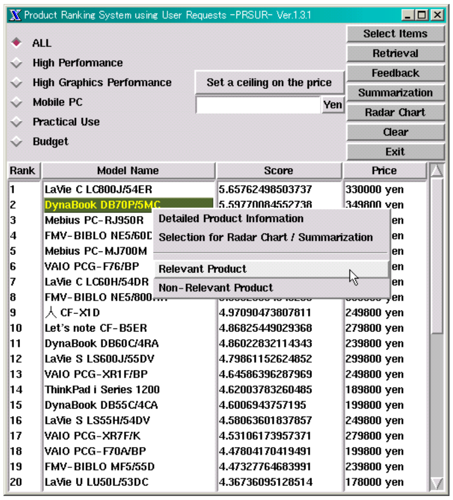
Figure 5 shows an example of table structure converted from HTML data (Fig. 3 (b)).

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


Product Ranking System using User Requests -PRSUR- Ver.1.3.1
Select Items
Retrieval
Feedback
Summarization
Radar Chart
Clear
Exit
Set a ceiling on the price
Yen
Detailed Product Information
Selection for Radar Chart / Summarization
Relevant Product
Non-Relevant Product

◆ ALL
◇ High Performance
◇ High Graphics Performance
◇ Mobile PC
◇ Practical Use
◇ Budget

Rank	Model Name	Score	Price
1	LaVie C LC800J/54ER	5.65762498503737	330000 yen
2	DynaBook DB70P/5MC	5.59770084552738	349800 yen
3	Mebius PC-RJ950R		en
4	FMV-BIBLO NE5/60D		en
5	Mebius PC-MJ700M		en
6	VAIO PCG-F76/BP		en
7	LaVie C LC60H/54DR		en
8	FMV-BIBLO NE5/800	[illegible]	en
9	人 CF-X1D	4.97090473807811	249800 yen
10	Let's note CF-B5ER	4.86825449029368	279800 yen
11	DynaBook DB60C/4RA	4.86022832114343	239800 yen
12	LaVie S LS600J/55DV	4.79861152624852	299800 yen
13	VAIO PCG-XR1F/BP	4.64586396287969	249800 yen
14	ThinkPad i Series 1200	4.62003783260485	189800 yen
15	DynaBook DB55C/4CA	4.6006943757195	199800 yen
16	LaVie S LS55H/54DV	4.58063601837857	249800 yen
17	VAIO PCG-XR7F/K	4.53106173957371	279800 yen
18	VAIO PCG-F70A/BP	4.47804170419491	199800 yen
19	FMV-BIBLO MF5/55D	4.47327764683991	239800 yen
20	LaVie U LU50L/53DC	4.36736095128514	178000 yen

Figure 6: A prototype system.

Table 1: Classification of Attributes.

Quantitative	Qualitative
CPU Clock:MHz, GHz	CPU Processor
Memory: MB	Graphics
Display: inch	CD-R/RW
Weight: kg	DVD-ROM, DVD-RAM
Dimensions: mm, cm	Pre-installed-OS
. . .	. . .

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 2: The classification of units for comparison (in the case of Japanese specifications).

Maximum-Best	MHz, MB, GB, inch
Minimum-Best	W, yen, $, Kg
Dependent on an attribute	hours, mm

Table 3: The classification of units for standard value.

Mean	W, yen, $, mm, hours, Kg
Mode	MHz, KB, MB, GB, colors, inch

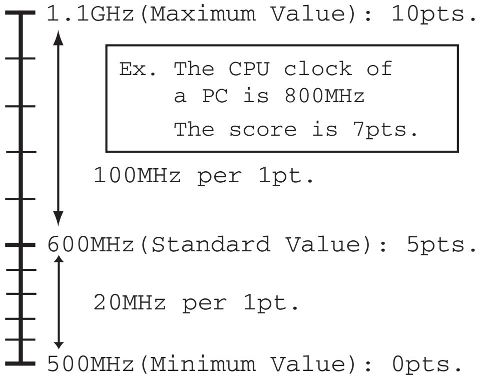

1.1GHz(Maximum Value): 10pts.
Ex. The CPU clock of a PC is 800MHz The score is 7pts.
100MHz per 1pt.
600MHz(Standard Value): 5pts.
20MHz per 1pt.
500MHz(Minimum Value): 0pts.

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:

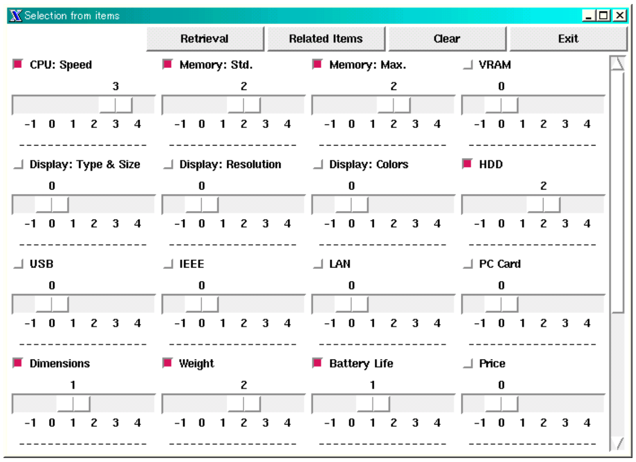
1. Search a keyword from all table structures with qualitative attributes.
2. If the keyword exists in all products, it is not extracted as the characteristic-data.
3. If the keywords possess weight, score the weight to the PC with the keywords (e.g., Pentium4: 4, Pentium3: 3, and Celeron: 2).
4. Extract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

Request	Attributes
High performance	CPU, Memory, Display, HDD, Interface
High graphics performance	Display, Graphics, CPU, Memory
Mobile PC	Battery life, Dimensions, Weight
Practical use	CPU, HDD, Price, Interface, Software
Budget PC	Price, Software, Memory, CPU



☒ CPU: Speed 3
☒ Memory: Std. 2
☒ Memory: Max. 2
□ VRAM 0
□ Display: Type & Size 0
□ Display: Resolution 0
□ Display: Colors 0
☒ HDD 2
□ USB 0
□ IEEE 0
□ LAN 0
□ PC Card 0
☒ Dimensions 1
☒ Weight 2
☒ Battery Life 1
□ Price 0

Figure 8: The window for attribute selection.

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

Attribute	Related attributes
CPU	Memory (Std. and Max.), HDD
Display	VRAM, Resolution, Colors
USB	IEEE, PC card
Weight	Dimensions, Battery life

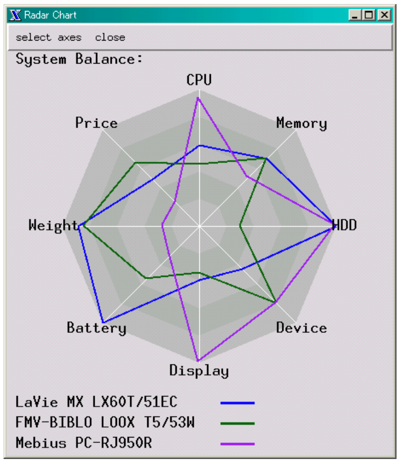


Figure 9: A radar chart.

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 $\mathbf{Q}_0 = (w(a_1, r), ..., w(a_n, r))$. A new query $\mathbf{Q}_1$ is computed as:

$$\mathbf{Q}_1 = \mathbf{Q}_0 + \alpha \sum_{i=1}^{N^+} \mathbf{D}_i^+ - \beta \sum_{i=1}^{N^-} \mathbf{D}_i^-$$

where $\mathbf{D}^+$ and $\mathbf{D}^-$ 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. α and β 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.

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

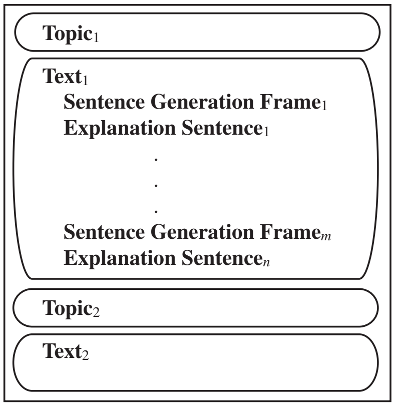


Figure 10: Document structure.

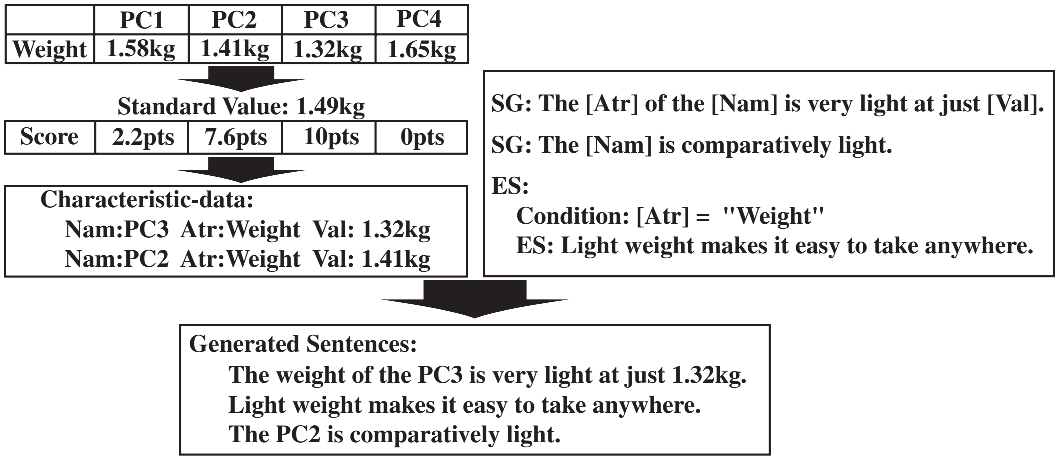
We define sentence generation frames (SG) and explanation sentences (ES) for sentence generation. Figure 10 shows document structure in our system. The number of topics is 9. Table 6 shows relationships between the topics and the attributes in specifications. Examples of SGs are as follows:

- Topic:
  [Topic] no {sugureta or yoi} [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 syuhenkiki 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.



	PC1	PC2	PC3	PC4
Weight	1.58kg	1.41kg	1.32kg	1.65kg

Standard Value: 1.49kg

Score	2.2pts	7.6pts	10pts	0pts

Characteristic-data:
Nam:PC3 Atr:Weight Val: 1.32kg
Nam:PC2 Atr:Weight Val: 1.41kg

SG: The [Atr] of the [Nam] is very light at just [Val].
SG: The [Nam] is comparatively light.
ES:
Condition: [Atr] = "Weight"
ES: Light weight makes it easy to take anywhere.

Generated Sentences:
The weight of the PC3 is very light at just 1.32kg.
Light weight makes it easy to take anywhere.
The PC2 is comparatively light.

Figure 11: Sentence generation processing.

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:

$$R = \frac{The\ Number\ of\ Documents\ Extracted\ Correctly}{Total\ Number\ of\ Correct\ Documents}$$

$$P = \frac{The\ Number\ of\ Documents\ Extracted\ Correctly}{Total\ Number\ of\ Extracted\ Documents}$$

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%.

		PC1	PC2	PC3
CPU		800MHz	700MHz	700MHz
Memory	Std.	64MB	64MB	64MB
	Max.	128MB	128MB	256MB
Hard Drive		15GB	15GB	10GB
Optical Device		external	external	internal
Display		11.3”TFT	12.1”TFT	10.4”TFT
VRAM		2.5MB	2MB	2MB
Resolution		1024×768pixels	800×600pixels	800×600pixels
PC card		TYPE II×1	TYPE II×1	TYPE II×1
Interface		USB×1, IrDA×1, LAN×1	USB×1, IrDA×1	USB×2
Key size		17mm	18mm	17mm
Battery Life	Std.	1.8hrs	2.5hrs	1.5hrs
	Max.	11hrs	5hrs	7hrs
Dimensions		270mm × 261mm × 29mm	270mm × 224mm × 33.7mm	257mm × 223mm × 29mm
Weight		1.6kg	1.98kg	1.5kg

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

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

Table 8: Correctness of characteristic-data

	Correctness
Eval(1)	319/365
Eval(2)	46/365

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.

Request	The ranking in the magazine	The ranking in our system
Request (1)	4th	3rd
Request (2)	8th	5th
Request (3)	1st	1st
Request (4)	4th	4th
Request (5)	1st	1st

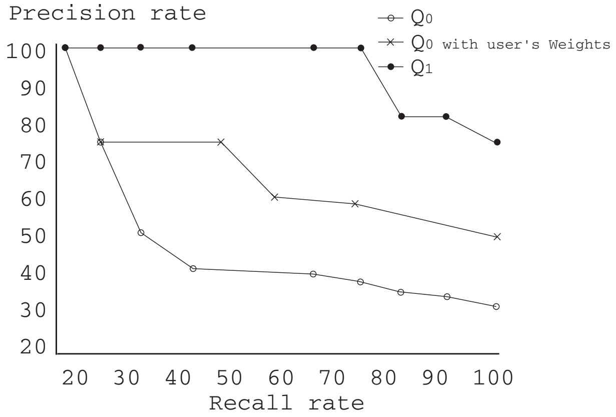


Figure 13: Recall-Precision rate.

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 $\mathbf{Q}_1$ obtained by user's feedback. "$\mathbf{Q}_0$ 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.

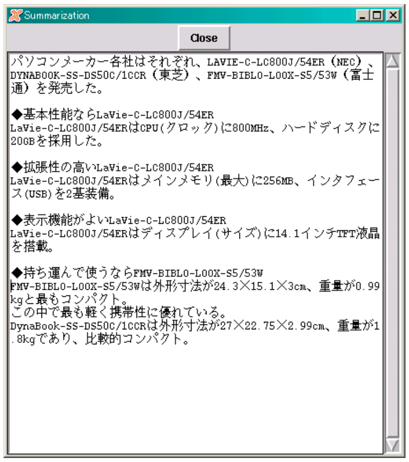
Eval(3) The redundancy of expression.

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.



パソコンメーカー各社はそれぞれ、LAVIE-C-LC800J/54ER（NEC）、DYNABOOK-SS-DS50C/1CCR（東芝）、FMV-BIBLO-LOOX-S5/53W（富士通）を発売した。

◆基本性能ならLaVie-C-LC800J/54ER
LaVie-C-LC800J/54ERはCPU(クロック)に800MHz、ハードディスクに20GBを採用した。

◆拡張性の高いLaVie-C-LC800J/54ER
LaVie-C-LC800J/54ERはメインメモリ(最大)に256MB、インタフェース(USB)を2基装備。

◆表示機能がよいLaVie-C-LC800J/54ER
LaVie-C-LC800J/54ERはディスプレイ(サイズ)に14.1インチTFT液晶を搭載。

◆持ち運んで使うならFMV-BIBLO-LOOX-S5/53W
FMV-BIBLO-LOOX-S5/53Wは外形寸法が24.3×15.1×3cm、重量が0.99kgと最もコンパクト。
この中で最も軽く携帯性に優れている。
DynaBook-SS-DS50C/1CCRは外形寸法が27×22.75×2.99cm、重量が1.8kgであり、比較的コンパクト。

Figure 14: Generated sentences.

Table 10: Evaluation of generated sentences.

Eval(1)	Eval(2)	Eval(3)	Eval(4)
3.7	3.7	4.3	4.0

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

[1] W. Cohen, "The Whirl approach to information integration", IEEE Intelligent SYSTEMS, Vol.13, No.5, pp.20–24, 1998.

[2] A. Fukumoto, K. Shimada, and Tsutomu Endo, "Information extraction from specifications on the World Wide Web," Proceedings of PACLING 2001, pp.109–116, 2001.

[3] T. Tokunaga, "Information Retrieval and Natural Language Processing," Computation and Language Volume 5, University of Tokyo Press, 1999 (in Japanese).

[4] Nikkei Best PC, Nikkei Business Publications, Inc.

[5] J. J. Rocchio, “Relevance feedback in information retrieval”, in The SMART Retrieval System: Experiments in Automatic Document Processing, Chapter 14, pp.313–323, Prentice-Hall, Inc., 1971.

[6] H. Matsuo and H. Kimoto, “A content extraction method from Japanese texts based on pattern matching using extraction patterns,” Trans. IPSJ, Vol.36, No.8, pp.1838–1844, 1995 (in Japanese).

[7] F. Masui and J. Fukumoto, “Information extraction for listing of product information,” Proceedings of Workshop Program The 4th Annual Meeting of The Association for Natural Language Processing, pp.56–63 (in Japanese).

[8] Y. Eriguchi and T. Kitani, “Information extraction from Japanese text using Tomita’s generalized LR parse,” Trans. IPSJ, Vol.38, No.1, pp.44–54, 1997 (in Japanese).

[9] S. Sekine, “Information extraction from text,” IPSJ Magazine, Vo.40, No.4, 1999 (in Japanese).

[10] T. Wakao, “Information extraction from English text,” Technical Report of IPSJ, NL114-12, pp.77–83, 1996 (in Japanese).

[11] J. Akamatsu, Y. Takao, H. Nagai, T. Nakamura, and H. Nomura, “Information extraction form newspaper articles of multiple products,” Technical Report of IPSJ, NL140-9, pp.61–68, 2000 (in Japanese).

[12] J. Hu, R. Kashi, D. Lopresti, and G. Wilfong, “Medium-independent table detection,” Proceedings of Document Recognition and Retrieval VII, pp.23–28, 2000.

[13] H.T. Ng, C.Y. Lim, and J.L.T. Koo, “Learning to recognize tables in free text,” Proceedings of the 37th Annual Meeting of ACL, pp.443–450, 1999.

[14] M. Sato, S. Sato, and Y. Shinoda, “Automatic digesting of the NetNews,” Trans. IPSJ, Vol.36, No.10, pp.2371–2379, 1995 (in Japanese).

[15] A. Kawai, T. Tsukamoto, K.Yamamoto, and T. Shiino, “Automatic extraction of relational information using document structure from itemization and tabular forms,” IEICE Trans. Inf. & Syst., Vol.J81-DII, No.7, pp.1609–1620, 1998 (in Japanese).

[16] H.H. Chen, S.C. Tsai, J.H. Tsai, “Mining tables from large scale HTML texts,” Proceedings of COLING2000, pp.166–172, 2000.

[17] J. Hammer, H. Garcia-Molina, J. Cho, R. Aranha, and A. Crespo, “Extracting semistructured information from the Web,” Proceedings of the Workshop on Management of Semistructured Data, 1997.

[18] M. Yoshida, K. Torisawa, and J. Tsujii, “Extracting ontologies from World Wide Web via HTML tables,” Proceedings of PACLING 2001, pp.332–341, 2001.

[19] M. Okumura and H. Nanba, “New topics on automated text summarization,” Journal of Natural Language Processing, Vol.9, No.4, pp.97–116, 2002.

[20] R. I. Kittredge and A. Polguere, “The generation of reports from databases,” In R. Dale, H. Moisl and H. Somers (eds.): A handbook of natural language processing, pp.261–304, 2000.

[21] D. Clark, “Shopbots become agents for business chance,” IEEE Computer, Vol.33, No.2, 2000.

[22] R.B. Doorenbos, O. Etzioni, and D.S. Weld “A scalable comparison-shopping agent for the World Wide Web,” Proceedings of the first International Conference on Autonomous Agents, 1997.

[23] J. Chai, V. Horvath, N. Kambhatla, N. Nicolov, and M. Budzikowska, “A conversational interface for online shopping,” Proceedings of HLT2001, 2001.

[24] M. Budzikowska, J. Chai, S. Govindappa, V. Horvath, and N. Kambhatla, “Conversational sales assistant for online shopping,” Proceedings of HLT2001, 2001.

[25] K. Shimada, K. Hayashi, and Tsutomu Endo, “Keywords and weighting for product specifications extraction,” Proceedings of PACLING2003, 2003.

[26] K. Shimada, T. Ito, and T. Endo, “Characteristic-data extraction - Re-evaluation by the specifications of two products -,” Technical Report of IEICE, TL2001-33, pp.27–34, 2001 (in Japanese).

[27] K. Shimada, T. Ito, T. Endo, “Classification of Images using Their Neighboring Sentences,” Proceedings of PACLING, pp.250-256, 2001.9.