KEYWORDS AND WEIGHTING FOR PRODUCT SPECIFICATIONS EXTRACTION

Kazutaka SHIMADA, Koji HAYASHI and Tsutomu ENDO
Department of Artificial Intelligence,
Kyushu Institute of Technology,
Iizuka, Fukuoka 820-8502 Japan
{*shimada, k_haya, endo*}*@pluto.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

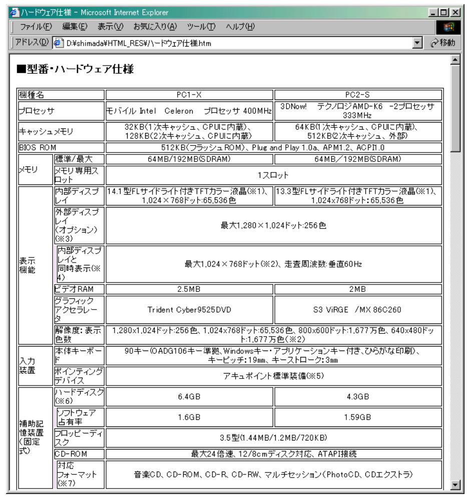
1. INTRODUCTION

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 process. 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:



ハードウェア仕様 - 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)

アドレス(D) D:¥shimada¥HTML_RES¥ハードウェア仕様.htm 移動

■型番・ハードウェア仕様

機種名		PC1-X	PC2-S
プロセッサ		モバイル Intel Celeron プロセッサ 400MHz	3DNow! テクノロジAMD-K6 -2プロセッサ 333MHz
キャッシュメモリ		32KB(1次キャッシュ、CPUに内蔵)、128KB(2次キャッシュ、CPUに内蔵)	64KB(1次キャッシュ、CPUに内蔵)、512KB(2次キャッシュ、外部)
BIOS ROM		512KB(フラッシュROM)、Plug and Play 1.0a、APM1.2、ACPI1.0	
メモリ	標準/最大	64MB/192MB(SDRAM)	64MB／192MB(SDRAM)
	メモリ専用スロット	1スロット	
表示機能	内部ディスプレイ	14.1型FLサイドライト付きTFTカラー液晶(※1)、1,024×768ドット:65,536色	13.3型FLサイドライト付きTFTカラー液晶(※1)、1,024x768ドット:65,536色
	外部ディスプレイ(オプション)(※3)	最大1,280×1,024ドット:256色	
	内部ディスプレイと同時表示(※4)	最大1,024×768ドット(※2)、走査周波数:垂直60Hz	
	ビデオRAM	2.5MB	2MB
	グラフィックアクセラレータ	Trident Cyber9525DVD	S3 ViRGE /MX 86C260
	解像度:表示色数	1,280x1,024ドット:256色、1,024x768ドット:65,536色、800x600ドット:1,677万色、640x480ドット:1,677万色(※2)	
入力装置	本体キーボード	90キー(OADG106キー準拠、Windowsキー・アプリケーションキー付き、ひらがな印刷)、キーピッチ:19mm、キーストローク:3mm	
	ポインティングデバイス	アキュポイント標準装備(※5)	
補助記憶装置(固定式)	ハードディスク(※6)	6.4GB	4.3GB
	ソフトウェア占有率	1.6GB	1.59GB
	フロッピーディスク	3.5型(1.44MB/1.2MB/720KB)	
	CD-ROM	最大24倍速、12/8cmディスク対応、ATAPI接続	
	対応フォーマット(※7)	音楽CD、CD-ROM、CD-R、CD-RW、マルチセッション(PhotoCD、CDエクストラ)	

Figure 1. Specifications of products.

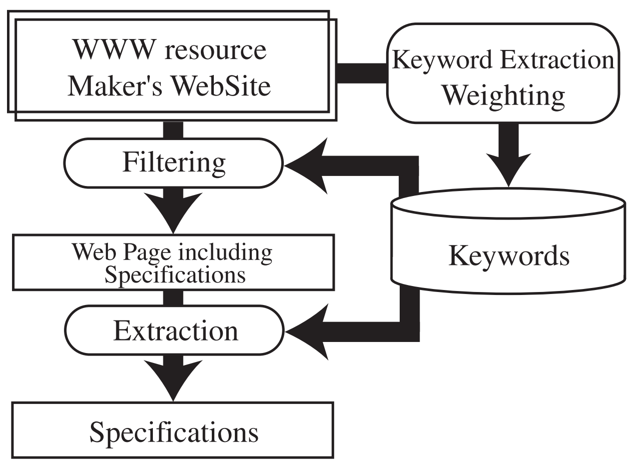


Figure 2. Outline of specification extraction (Table Detection in Fig 3.)

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.

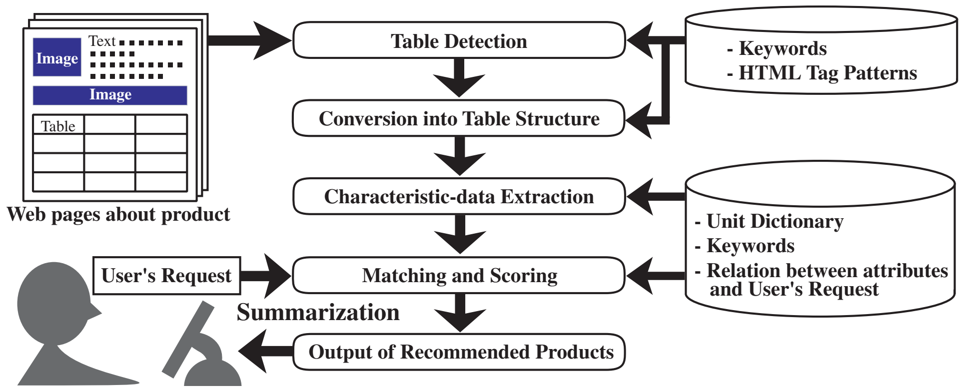


FIGURE 3. Outline of our system.

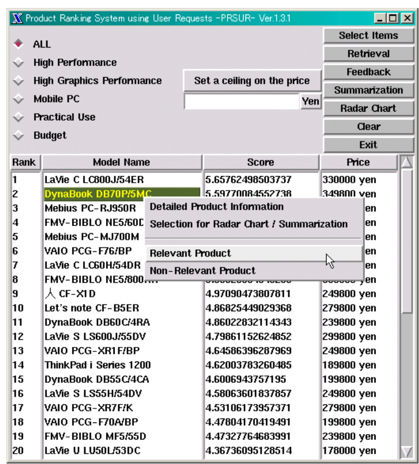


FIGURE 4. A prototype system.

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 $term_t$ in D_{real} and the weight of $term_t$ in D_{no} are computed as:

$$wr_t^{real} = \frac{w_t^{D_{real}}}{w_t^{D_{no}}}, \quad wr_t^{no} = \frac{w_t^{D_{no}}}{w_t^{D_{real}}},$$

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}. The weight of $term_t$ as KW is $ws_t^{real} = df(t, D_{real}) \times wr_t^{real}$. The weight of $term_t$ as NW is $ws_t^{no} = df(t, D_{no}) \times wr_t^{no}$. $df(t, D_{real})$ and $df(t, D_{no})$ are the number of documents including $term_t$ in D_{real} and D_{no} respectively. We employ the words of top ranks of ws_t^{real} and ws_t^{no} 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 $term_t$ belongs to class C_i is given by:

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

where $C = \{D_{real}, D_{no}\}$. We handle the probabilities as the weights of KWs and NWs. In other words, the ws_t^{real} is $P(D_{real}|t)$ and the w_t^{no} is $P(D_{no}|t)$. The conditions of the KWs are $P(D_{real}|t) \geq 0.75$ and $df(t, D_{real}) \geq \frac{D_{real}^M}{2}$. The conditions of the NWs are $P(D_{no}|t) \geq 0.75$ and $df(t, D_{no}) \geq \frac{D_{no}^M}{10}$. D_{real}^M and D_{no}^M are the number of documents in D_{real} and D_{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 <TABLE> tag from an HTML document d_i.
2. Extract the contents ($Cont$) of <TD> tags in the <TABLE> tag.
3. Compute

$$Ratio_{real} = \frac{\sum_{t \in Cont} ws_t^{real}}{\sum_{t \in KW} ws_t^{real}}, \quad Ratio_{no} = \frac{\sum_{t \in Cont} ws_t^{no}}{\sum_{t \in NW} ws_t^{no}} .$$

4. Compute

$$Score_i = Ratio_{real} \times \frac{Ratio_{real}}{Ratio_{no}} .$$

5. If $Score_i$ is more than or equal to a threshold $th1$, extract the HTML document d_i.

TABLE 1. Dataset

Product		Training	Test
PC	DocIncSpecs	50	100
	DocNotIncSpecs	50	100
Digital Camera	DocIncSpecs	50	93
	DocNotIncSpecs	50	100

TABLE 2. Keyword and Noise-word

Product		Keyword	Noise-word
PC	Entropy	30	15
	Bayes	19	15
Digital Camera	Entropy	30	15
	Bayes	28	22

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 $Sum_i = \sum_{t \in Cont} ws_t^{real}$ for $table_i$.
5. Extract $table_i$ maximizing Sum_i.
6. If Sum_i is more than or equal to a threshold $th2$, extract $table_i$ as specifications. $th2$ is $\frac{\sum_{t \in KW} ws_t^{real}}{2}$.

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 DocNotIncSpecs 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:

$$Recall\ \ (R) = \frac{\#\ \ of\ \ extracted\ \ correct\ \ documents}{\#\ \ of\ \ documents\ \ including\ \ specifications}$$

KWs by entropy	KWs by Bayes
surotto (slot) :: 272507.986 naizou (internal) :: 149048.428 gaibu (external) :: 147305.761	kyasshu (cache) :: 1 saundo (sound) :: 1 surotto (slot) :: 0.97361 ....

NWs by entropy	NWs by Bayes
nichi (day) :: 13631.914 tsuki (month) :: 13631.912 gazou (image) :: 10515.892	appu (up) :: 0.95219 saito (site) :: 0.760247 syoppingu (shopping) :: 0.796453 ....

FIGURE 5. Examples of KWs and NWs for PCs.

KWs by entropy	KWs by Bayes
rosyutsu (exposure) :: 195626.168 syatta (shutter) :: 194430.843 howaito (white) :: 185551.034	douga (moving picture) :: 1 howaito (white) :: 1 rosyutsu (exposure) :: 0.9705228 ....

NWs by entropy	NWs by Bayes
mei (name) :: 16105.166 fuirumu (film) :: 15575.736 kitto (kits) :: 12905.866	kitto (kits) :: 1 mei (name) :: 0.97323043 fuirumu (film) :: 0.9704957 ....

FIGURE 6. Examples of KWs and NWs for digital still cameras.

$$Precision \quad (P) = \frac{\# \;\; of \;\; extracted \;\; correct \;\; documents}{\# \;\; of \;\; extracted \;\; documents}$$

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

For a filtering process, we set α 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-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:

$$ws_t^{real} = \frac{P(D_{real}|t)}{P(D_{no}|t)}, \qquad ws_t^{no} = \frac{P(D_{no}|t)}{P(D_{real}|t)}.$$

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

TABLE 3. Filtering

Product	Method	*th*1	*Recall*	*Precision*	*F-measure*
PC	Entropy	0.25	100.0%	81.3%	91.6%
	Bayes	0.50	100.0%	99.0%	99.6%
Digital Camera	Entropy	0.80	94.6%	75.9%	86.1%
	Bayes	0.55	96.8%	94.7%	96.0%

TABLE 4. Extraction

Product	Method	*th*1	*Recall*	*Precision*
PC	Entropy	0.25	93.0%	96.9%
	Bayes	0.50	95.0%	100.0%
Digital Camera	Entropy	0.80	80.7%	86.6%
	Bayes	0.55	82.8%	95.1%

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

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

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

[3] Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Hirano, H. Matsuda, and M. Asahara, "Japanese morphological analysis system ChaSen version 2.0 manual 2nd edition," 1999.

[4] P. Jackson and I. Moulinier, "Natural language processing for online applications," Natural Language Processing, Volume 5, John Benjamins Publishing Company, 2002.

[5] K. Shimada, A. Fukumoto, and T. Endo, "Information extraction from personal computer specifications on the Web using a user's request," IEICE Transactions on Information and Systems, vol.E86-D, no.8, Aug. 2003.

[6] K. Shimada, T. Ito, and T. Endo, "Multiform Summarization from Product Specifications," Proceedings of PACLING 2003.

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

[8] Y. Wang and J. Hu, "A machine learning based approach for table detection on the Web," Proc. of The Eleventh International World Web Conference, 2002.

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