Acquisition of new training data from unlabeled data for product specification extraction

Kazutaka Shimada and Tsutomu Endo
Department of Artificial Intelligence
Kyushu Institute of Technology 680-4 Iizuka Fukuoka 820-8502 Japan
{shimada, endo}@pluto.ai.kyutech.ac.jp

Abstract

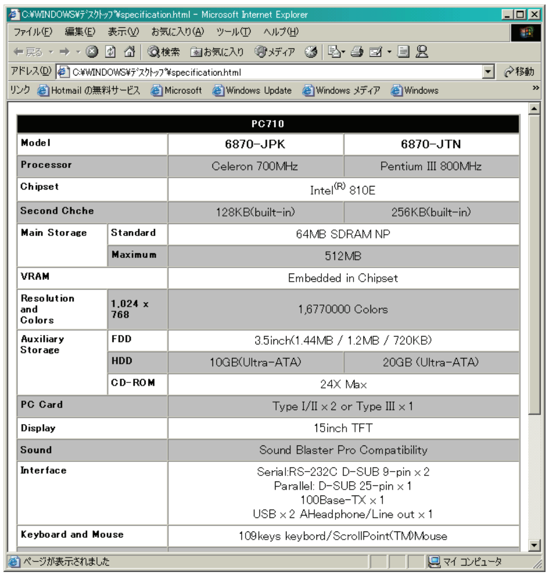
In this paper we describe a method for extracting product information. In general, information about products, such as specifications, is written in tabular form. We focus on product specifications. Specification extraction is a significant task to treat product information. This task is the basis of various systems, such as construction of a product database. A problem in this task concerns the size of positive data in training data. The positive data denotes documents including target information (product specifications in this case). It is approximately 2-5% in training data. With regard to machine learning methods, lack of data generally generates a weak classifier. Hence we need to increase the amount of positive data automatically. To solve this problem, we propose a method for extracting new positive data using a similarity measure. We improve the accuracy of a classifier using original training data and new positive data extracted from unlabeled data. Experimental results show the effectiveness of our method.

Keywords : Table, Product specification, Infornation extraction, Similarity

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. Information on the WWW is in the form of not only text but also images and tables. Although tables are structured information, i.e., attribute-value pairs, most of conventional information retrieval systems treat tables as text. Since tables are an efficient way to express relational information, table extraction is a significant task for web mining, QA systems, summarization and so on [4, 5, 9, 11, 13].

Here we consider product information. Specifications about the equipment for products, such as personal computers and digital still cameras, are one of the useful online documents. The specifications are usually utilized on online-shopping sites.


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.

We focus on extraction of product specifications. We have also developed a multi-specifications summarization system for decision support for shopping [11]. 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. However, 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]. The systems that handle product information need to extract correctly specifications written in tabular form.

We have reported methods using support vector machines (SVMs) and transductive SVMs for this task [2]. A problem in this task concerns the number of positive data for machine learning methods. The positive data denotes documents including target information[1]. It is approximately 2-5% in all data. In the case that the number of positive data is insufficient, a low-accuracy classifier is generated. To generate a high-accuracy classifier,

[1]In this paper it is product specifications.

machine learning methods require a large amount of training data. However, collecting the training data by hand is costly. To solve this problem, we need to increase the amount of positive data automatically. In this paper, we propose a method for extracting new training data from an unlabeled data set using a similarity measure. We improve the performance of a classifier using a small training data set and the extracted data. We evaluate the effectiveness of the proposed method by applying to documents of specifications of PCs, digital still cameras and printers. We compare SVMs and TSVMs using our method with original SVMs and TSVMs.

2 Related Work

This paper focuses on product specification extraction, namely extraction of particular tables. There are several approaches to extract information using document structures such as itemization and tabular form. Traditionally, they handle a document image or plain text [3, 8, 9].

On the other hand, there are several approaches to treat HTML-based tables. Itai et al. have reported a method for information extraction from HTML pages and integration [5]. They did not, however, discuss a method for extracting tables. Although Chen et al. have reported a method for extracting tables from HTML documents, they employed heuristic rules for table extraction [1]. Constructing rules by handwork is costly. We have reported a method for table extraction using Bayes' rule [10]. Wang et al. have evaluated a table extraction task by machine learning based approaches: decision tree learning and SVMs [14]. These studies did not, however, discuss the reduction of the training data. Yoshida et al. have proposed a method for table structure recognition using an unsupervised method that was achieved by utilizing the EM algorithm [15]. However, the accuracy was lower than the Wang's method, i.e., SVMs. Transductive SVMs (TSVMs) are a method to obtain high accuracy with a small training dataset and unlabeled data. TSVMs have been used for text classification, and have improved the performance of SVMs [6]. In this paper we improve the performance of SVMs and TSVMs by using our method.

3 Classifiers and Feature Selection

In this section, we describe classifiers for product specification extraction and a method of feature selection for their classifier. We use SVMs and transductive SVMs as the classifiers.

3.1 SVMs

SVMs are a machine learning algorithm that was introduced by [12]. They have been applied to tasks such as face recognition and text classification. An SVM is a binary classifier that finds a maximal margin separating hyperplane between two classes. The hyperplane can be written as:

$$y_i = \vec{w} \cdot \vec{x} + b$$

where $\vec{x}$ is an arbitrary data point, i.e., feature vectors, $\vec{w}$ and b are decided by optimization, and $y_i \in \{+1, -1\}$. The instances that lie closest to the hyperplane are called support vectors. Figure 2 (a) shows an example of the hyperplane. In the figure, solid line shows hyperplane $\vec{w} \cdot \vec{x} + b = 0$.

3.2 Transductive SVM

SVMs are a powerful approach for solving various classification problems. However, collecting the training data for machine learning methods is costly. SVMs can perform transduction by finding the hyperplane that maximizes the margin relative to both the labeled and unlabeled data. This solves the problem of the cost of collecting the training data. Transductive SVMs that were introduced by Joachims [6] take into account a particular test set and try to minimize misclassifications of those particular examples. They improve the performance of SVMs. The algorithm is as follows:

1. Train an inductive SVM on the training data.
2. Classify the test data by using the hyperplane.
3. Output the predicted classification of the test data, on the basis of the distribution of the positive and negative data.
4. Identify the two examples which lead to the reduction of misclassifications by changing the class labels of them.
5. Switch these examples and retrain them.

Figure 2 (b) shows an example of TSVMs. The dotted line in the figure is the hyperplane of an inductive SVM. Solid circles are unlabeled examples. The solid line shows the transductive classification. For our experiment, we use the SVM^{light} system implemented by Thorsten Joachims[2]. We use a linear kernel for both SVMs and TSVMs in our experiment.

3.3 Feature Selection

Our system extracts keywords as feature vectors for SVMs. Here we utilize the structure of specifications written in tabular form for the keyword

[2]http://svmlight.joachims.org/

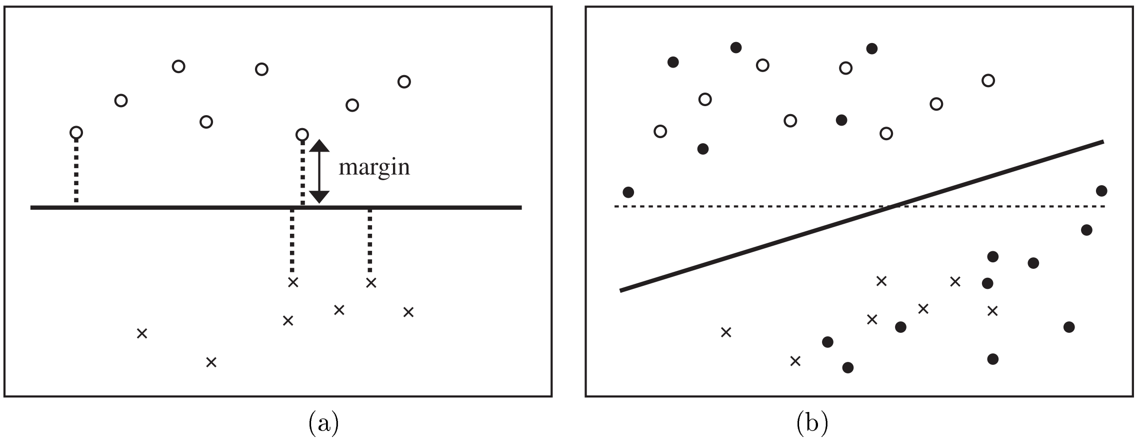


Figure 2: SVMs and transductive SVMs.

extraction. The structure is that, generally, attributes in specifications correspond to the 1st column in tables and values that belong to the attributes correspond to the rest of columns. Requirements for keyword extraction are as follows:

1. Words in the 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 use words that satisfy these three requirements.

We handle the contents of the 1st <TD> in <TR> tags. The reason why we use only the 1st column for the keyword extraction is that words in attributes are rarely changed to other words. For example, the word "CPU" appears in attributes about most of PCs specifications. However, the values of the attribute "CPU" are often changed (Pentium $\leftrightarrow$ Celeron, GHz $\leftrightarrow$ MHz). Therefore words in attributes are characteristic words for detection of specifications. 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 [7].

Next, the system computes the weights of the keyword candidates and then extracts keywords from them, i.e., the feature selection for SVMs and TSVMs.

In this paper, we use normalized $tf \cdot idf$ for weighting. A keyword has two indicators: ws^{pos} and ws^{neg}. The ws^{pos} is the weight of a keyword in D_{pos}. If a keyword appears frequently in documents that contain specifications and appears infrequently in documents that do not contain specifications, the value of ws^{pos} is high. The ws^{neg} is the weight of a keyword in D_{neg}. D_{pos} and D_{neg} are documents including specifications and documents not including specifications respectively.

normalized $tf \cdot idf$

$tf \cdot idf$ is one of the most popular method for weighting. The $tf \cdot idf$ weight of a term in one document is the term frequency (tf) divided by its document frequency (idf). In this paper, we employ a method derived from the normal $tf \cdot idf$ with some adjustments. This method was used for a table detection task by Wang et al [14]. The $tf \cdot idf$ weight of $term_t$ in D_{real} and the weight of $term_t$ in D_{no} are computed as:

$$w_t^{pos} = \sum_{d_i \in D_{pos}} tf(t, d_i) \times \log\left(\frac{df_t^{pos}}{N_{pos}} \frac{N_{neg}}{df_t^{neg}} + 1\right)$$

$$w_t^{neg} = \sum_{d_i \in D_{neg}} tf(t, d_i) \times \log\left(\frac{df_t^{neg}}{N_{neg}} \frac{N_{pos}}{df_t^{pos}} + 1\right)$$

where N_{pos} and N_{neg} are the number of documents in D_{pos} and D_{neg} respectively. df_t^{pos} and df_t^{neg} are the number of documents including $term_t$ in N_{pos} and N_{neg} respectively. We use cosine normalization for weighting.

$$ws_t^{pos} = \frac{w_t^{pos}}{Norm_{pos}}, \quad ws_t^{neg} = \frac{w_t^{neg}}{Norm_{neg}}$$

where

$$Norm_{pos} = \sqrt{\sum_{t \in D_{pos}} w_t^{pos} \times w_t^{pos}},$$

$$Norm_{neg} = \sqrt{\sum_{t \in D_{neg}} w_t^{no} \times w_t^{neg}}$$

We use the words exceeding a threshold. These words are feature vectors for SVMs. The weight of a word is the value of the vector (the word).

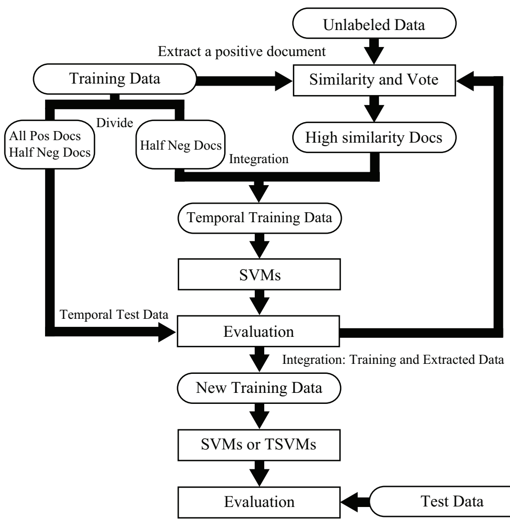


Figure 3: Outline of our process.

4 Extraction of Positive Data from Unlabeled Data

For a small number of training data, the size of positive data in them is a problem in this task. It is approximately 2-5% in all data. In the case that the number of positive data is insufficient, machine learning methods usually generate a low-accuracy classifier. Therefore we need to extract more amount of positive data from unlabeled data automatically.

In this paper, we compute the degree of similarity between a positive document in training data and documents in unlabeled data. Figure 3 shows the process flow of the proposed method. Our method determines an appropriate threshold in the extraction process automatically. The algorithm is as follows:

(1) Extract a positive document from training data.

(2) Compute the similarity between the positive document and documents in unlabeled data. The similarity is computed as follows:

$$sim(p_x, u_y) = \frac{\sum_{i=1}^{N} x_i \cdot y_i}{\sqrt{\sum_{i=1}^{N} x_i^2 \times \sum_{i=1}^{N} y_i^2}}$$

where p and u are a positive document and a document in unlabeled data respectively. x_i and y_i are the value of a word i in p and u respectively. These words and values are the same as feature vectors for SVMs. N is the number of words (vectors). This formula is well-known as the cosine measure.

(3) If the similarity is more than or equal to a threshold th_1, vote for the document.
For the documents voted in (3),

Table 1: Dataset

	PC	Digital Camera	Printer
Specs	239	117	143
NotSpecs	4761	4883	4857

(3-1) Compute the similarity between the documents and the other documents in the unlabeled data.

(3-2) If the similarity is more than or equal to a threshold th_1, vote for the document.

(4) If the number of votes obtained is more than or equal to a threshold th_2, extract the document as positive data.

(5) Divide the training data into two groups: (a) temporal training data and (b) temporal test data.
The temporal training data consists of the extracted documents and half of negative documents in the training data. The temporal test data consists of the other documents in the training data.

(6) Generate a classifier from the temporal training data (a). As regards the classifier, we use SVMs.

(7) Evaluate the performance of the classifier with the temporal test data (b).

(8) Decrease th_1 and go back to **(2)** in the case that th_1 is higher than a limit.

(9) Compare the performance of the classifiers in each turn: the results in **(7)**. Detect the appropriate th_1.

(10) Integrate original training data and the data extracted by the appropriate th_1.

Determination of the appropriate th_1 is an issue in this proposed method. In this paper, we search for the th_1 in which the precision rate decreases as compared with the previous turn. In other words, the process is to identify a disimproved turn in **(7)**. Here we define the th_1 of the previous turn as the appropriate th_1 because the decrease of the precision rate in this method denotes that the method extracted negative documents as positive documents.

5 Evaluation

We evaluated the appropriateness for the proposed method with a dataset. The data was extracted from 28 manufacturer's sites by a file downloading software. The data consists of web documents about PCs, digital still cameras and printers. The total number of pages is 86737. The dataset for

Table 2: Experimental result

Product	Measure	(1) SVM	(2) Our Method + SVM	(3) TSVM	(4) Our Method + TSVM
PC	P	**0.9171**	0.8438	0.8384	0.8360
	R	0.6928	0.8662	0.8789	**0.9112**
	F	0.7711	0.8363	0.8551	**0.8708**
Digital Camera	P	**0.9582**	0.8613	0.7264	0.7218
	R	0.5201	0.8428	0.8348	**0.8840**
	F	0.6408	**0.8391**	0.7608	0.7762
Printer	P	**0.8944**	0.7971	0.7670	0.8402
	R	0.4798	**0.8612**	0.7891	0.8266
	F	0.5734	**0.8247**	0.7574	0.8179

evaluation was constructed by 5000 web documents extracted from all the data randomly. These documents were divided into 100 documents as training data and 4,900 documents as test data. Unlabeled data for our method and TSVMs was 1000 documents in the test data. We evaluated this experiment with 5 training data extracted from the 5000 documents. The number of data is shown in Table 1. In Table 1, the Specs and the NotSpecs denote documents including specifications and documents not including specifications, respectively. The NotSpecs includes specifications of other products. For example, the NotSpecs for digital cameras includes the specifications of video cameras and still cameras.

We evaluated this task by *F-measure*. The *F-measure* is computed as follows:

$$F\text{-}measure \quad (F) = \frac{1}{\frac{1}{2P} + \frac{1}{2R}}$$

where R and P denote a *Recall* and a *Precision* rate respectively.

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

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

We set the initial th_1 to 0.95. The th_1 was decreased in decrements of 0.05 until 0.35. We set the th_2 to $vote\_num \times \frac{2}{3}$. The $vote\_num$ denotes the number of votes obtained in the step **(3)** in Sect. 4.

We compared 4 methods: (1) SVMs, (2) SVMs with our method, (3) TSVMs, and (4) TSVMs with our method. The experimental results are shown in Table 2. For PCs, the method (4), TSVMs with our method, produced the best performance on *F-measure*. For digital cameras and printers, the method (2), SVMs with our method, yielded the best *F-measure*. Our method and TSVMs improved the recall rate, as compared with SVMs. This led to the improvement of *F-measure*. The recall rates of SVMs was insufficient although the precision rates of SVMs were higher than those of our method and TSVMs. The results show the effectiveness of the proposed method.

Table 3: The result of acquisition of new training data

Product	POS	AVE	R	P
PCs	45	6.8	0.1511	0.8909
Cameras	16	4.2	0.2625	0.5899
Printers	34	5	0.1471	0.6950

To accomplish higher *F-measure*, we need further consideration for the determination of two thresholds th_1 and th_2. Table 3 shows recall and precision rates of the acquisition process of positive data from unlabeled data. The POS in Table 3 is the number of positive data (documents including specifications) in 1000 unlabeled data. The AVE in Table 3 is the average of the number of documents extracted correctly in the process. The results denote that our method often extracted negative documents as positive documents. In this process, the precision rate is the most important factor. If the process extracts negative documents as positive documents, it generates a low-accuracy classifier because the new training data contains incorrect labeled data. In particular, the precision rates of digital cameras and printers were insufficient. The results caused the decline of the precision rates in the product specification extraction (Table 2). Therefore, the estimation of the appropriate thresholds is one of the most important task. Also, we need to treat other similarity measures for the improvement of accuracy.

6 Conclusions

Table extraction in web documents is an interesting problem with many applications, such as web mining, QA systems and summarization. In this paper, we focused on product specifications presented in tabular form. A problem in this task concerns the number of positive data for machine learning methods. We proposed a method for extraction of new training data using a similarity measure. SVMs and TSVMs with our method outperformed original SVMs. Experimental results show the effectiveness of this method. The accuracy of the acquisition process of positive data from unlabeled data was insufficient. To accomplish higher *F-measure*, we need further consideration for the determination of two thresholds in the proposed method. Our future work includes (1) evaluation with other machine learning methods, (2) estimation of appropriate thresholds, and (3) consideration of other similarity measures.

References

[1] H.H. Chen, S.C. Tsai, and J.H. Tsai. Mining tables from large scale html texts. In *Proceedings of COLING2000*, pages 166–172, 2000.

[2] K. Hayashi, K. Shimada, and T. Endo. Product specification extraction using machine learning methods. In *Proceedings of the 10th Annual Meeting of The Association for Natural Language Processing (in Japanese)*, pages 733–736, 2004.

[3] J. Hu, R. Kashi, D. Lopresti, and G. Wilfong. Medium-independent table detection. In *Proceedings of Document Recognition and Retrieval VII*, pages 23–28, 2000.

[4] M. Hurst. Layout and language: Challenges for table understanding on the web. In *Proceedings of Workshop on Web Document Analysis, WDA01*, pages 27–30, 2001.

[5] K. Itai, A. Takasu, and J. Adachi. Information extraction from html pages and its integration. In *Proceedings of the 2003 Symposium on Application and the Internet Workshops (SAINT03)*, pages 276–281, 2003.

[6] T. Joachims. Transductive inference for text classification using support vecor machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 200–209, 1999.

[7] Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Hirano, H. Matsuda, , and M. Asahara. Japanese morphological analysis system chasen version 2.0 manual 2nd edition. Technical report, 1999.

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

[9] D. Pinto, A. McCallum, X. Wei, and W. B. Croft. Table extraction using conditional random feilds. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 235–242, 2003.

[10] K. Shimada, K. Hayashi, and T. Endo. Keyword and weighting for product specifications extraction. In *Proceedings of PACLING 2003*, pages 285–293, 2003.

[11] K. Shimada, T. Ito, and T. Endo. Multiform summarization from product specifications. In *Proceedings of PACLING 2003*, pages 83–92, 2003.

[12] V. N. Vapnik. *Statistical Learning Theory.* Wiley, 1999.

[13] H. L. Wang, S. H. Wu, K. K. Wang, C. L. Sung, W. L. Hsu, and W. K. Shih. Semantic search on internet tabular information extraction for answering queries. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 243–249, 2000.

[14] Y. Wang and J. Hu. A machine learning based approach for table detection on the web. In *Proceedings of The Eleventh International World Web Conference*, 2002.

[15] M. Yoshida, K. Torisawa, and J. Tsujii. Extracting ontologies from world wide web via html tables. In *Proceedings of PACLING 2001*, pages 332–341, 2001.