Proposal of a Spreadsheet-Based Search Tool for Cross-Searching Product Information of Japan's Shopping Websites

Somchai Chatvichienchai and Yamato Itakura

***Abstract*—In Japan, price comparison websites have become popular for finding and comparing the products from leading shopping websites. However, there are two drawbacks in product search of the price comparison websites. Firstly, customers cannot find new products of shopping websites that have not yet registered by the price comparison websites. Lastly, some product information of the price comparison websites is not up to date. In these situations, the customers need to search product information on shopping websites. This task is very time consuming. The objective of this research is to propose xSearchPro, a search tool that employs spreadsheet software as a user interface in performing cross-searching on Japan's shopping websites. The cross-searching is realized by using product search APIs (Application Program Interfaces) of the shopping websites. This paper also presents a method that automatically extracts some product information from product webpages of shopping websites. xSearchPro outputs search results in a tabular format to enable efficient product comparison.**

***Index Terms*—Search, user interface, spreadsheet, category, XML, JSON, website**

I. Introduction

Online shopping has overtaken traditional store shopping in popularity. According to a domestic survey report of Sumitomo Mitsui Card [1], 78.4% of questionnaire respondents used online shopping websites from April to July 2020, and 35.6% of all respondents said they used online shopping websites more frequently than in 2019. In Japan, price comparison websites (e.g., kakaku.com[1], hikaku.com[2], etc.) have become popular for finding and comparing the products that customers want and enjoy the cheapest prices on a wide range of top brand products from leading shopping websites such as Amazon[3], Rakuten Ichiba[4], etc. The price comparison websites need to periodically collect product information from shopping websites and to record the categorized product information into their databases. This causes the following two common drawbacks in searching products of the price comparison websites. Firstly, customers cannot find the new products of shopping websites that have not yet recorded into the databases. Lastly, some product information of the price comparison websites is not up to date. For example, a customer may find a USB flash memory that meets her requirement in a price comparison website. However, after clicking URL link of shopping site, she found that the USB flash memory of the shopping site has already sold out. Many customers may tackle the above two drawbacks by directly searching products on shopping websites. However, they need to screen each product based on all the provided information carefully in the ranking list, which is time-consuming. They need a better product search tool. Information retrieval research to date has been focused on optimizing search ranking algorithms for web documents while little attention has been paid to product search [2]. There are several intrinsic differences between web search and product search that make the direct application of traditional search ranking algorithms to E-Commerce search platforms difficult [3]. Previous work on product search aimed to enhance product search engines by proposing new ranking models [4], [5] that predict best-selling products. However, the previous work has failed to satisfy the demand of customers who are interested in the products other than the best-selling ones.

The objective of this research is to propose a product search tool called xSearchPro to solve the above two drawbacks. We first focus on cross-searching product information of Rakuten Ichiba website and Yahoo! Shopping[5] website since both websites allow system developers to use their product search APIs without charge under some agreements. However, there are some data (such as item number, additional shipping cost, etc.) that cannot obtain by the APIs. To handle this problem, we developed a method that automatically extracts the required data from the product web pages. To save manpower and time in developing xSearchPro, we developed xSearchPro by using macro feature of Microsoft Excel that provides sufficient APIs for developing web client applications. Since product search result is outputted to an Excel worksheet, the users can arbitrarily use filter and sort functions of Excel to arrange the search result into a pattern that helps them finding the required products efficiently. To the best of our knowledge, there is no previous research that has proposed a search tool like ours.

The organization of the paper is as follows. Section 2 presents the issues in developing xSearchPro. In section 3, we explain system design of xSearchPro. Section 4 presents comparison response time of xSearchPro with that of manual operation. In section 5, we discuss related work. The last section concludes this work and future work.

II. Issues in Developing xSearchPro

Issue 1: How Does xSearchPro Send Search Request to Shopping Websites and Obtain Search Result?

We employ the following APIs to realize the above functions.

Manuscript received March 19, 2021; revised October 5, 2021.

Somchai Chatvichienchai is with Department of Information Security, University of Nagasaki, 851-2195 Japan (e-mail: somchaic@sun.ac.jp).

Yamato Itakura was with Department of Information Security, University of Nagasaki. (e-mail: bs217003@sun.ac.jp).

[1] https://kakaku.com/
[2] https://www.hikaku.com/
[3] https://www.amazon.co.jp/
[4] https://www.rakuten.co.jp//
[5] https://shopping.yahoo.co.jp/

doi: 10.18178/jacn.2021.9.2.281

- Rakuten Product Search API [6]
- Rakuten Product Category Search API [7]
- API for searching products in Yahoo! Shopping [8].
- API for obtaining product category information of Yahoo! Shopping [9]

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <Items>
    <Item>
      <smallImageUrls>
        <imageUrl>https://thumbnail.image.rakuten.co.jp/@0_mall/excellar/cabinet/image0271/1000015471_1.jpg?_ex=64x64</imageUrl>
      </smallImageUrls>
……..
      <itemCaption>BUFFALO USB3.1 Push Slide USB Memory</itemCaption>
      <catchcopy> RUF3-SP64G-BK [64GB Black] </catchcopy>
……..
      <shopUrl>https://www.rakuten.co.jp/excellar/</shopUrl>
……..
      <itemCode>excellar:13351570</itemCode>
      <postageFlag>0</postageFlag>
      <itemName>64GB USB Memory USB3.1 Gen1-A/Type-C … SDDDC3-064G-G46</itemName>
      <itemPrice>1099</itemPrice>
      <shopName>excellar</shopName>
      <reviewCount>6</reviewCount>
      ……..
    </Item>
      ……..
  <Items>
</root>
```

Fig. 1. An example of a part of product information of Rakuten Ichiba described in XML format.

These APIs enable xSearchPro to perform product search by keywords, JAN code, ISBN code, category ID, brand ID, and store ID, and then to obtain product information. The product information answered by the APIs is described by XML [10] or JSON [11] for Rakuten Ichiba and JSON for Yahoo! Shopping, respectively. Due to space limitation of the paper, we will briefly explain Rakuten Item Search API. The API allows definition of output parameters as shown in (1), e.g., searching by item code, searching by existence of reviews, searching by expected delivery date, etc.

https://app.rakuten.co.jp/services/api/IchibaItem/Search/20170706?[parameter]=[value]… (1)

The values of *keyword* and *sort* must be encoded in UTF-8 style. Note that the whole request does not need to be encoded, only the individual value portions of it. For example, a keyword search for "USB memory 128GB" that has been ordered with the cheapest item first (sort=+itemPrice) will look like the URL shown in (2).

[APPLICATION ID] denotes developer identification, which an application developer must registered before using the APIs. Fig. 1 shows a part of an example of product information obtained after requesting product search shown in (2).

Fig. 2. An example of product information shown at Rakuten Ichiba website.

Fig. 3. An example of direction of sight movement of a user when comparing product information of Rakuten Ichiba website.

https://app.rakuten.co.jp/services/api/IchibaItem/Search/20170706?format=xml
&keyword=USB%20memory%20128GB
&sort=%2BitemPrice
&applicationId=[APPLICATION ID] (2)

Issue 2: How does xSearchPro enable the users comparing products immediately?

Fig. 2 depicts an example of information of a product outputted at Rakuten Ichiba website. Note that Yahoo! Shopping website also presents product information in the same style. To decide which product to buy, the users need to look through the product information and direct their gaze in the direction indicated by the arrow in Fig. 3. The more product that the users examine in this way, the more stressed they will feel. To handle this issue, xSearchPro will output search result to an Excel worksheet shown in Fig. 4. This output style of Fig. 4 can eliminate time and effort of the users on searching other data (such as *average review score*, *stock availability*, etc.) from product detail web page. In case the users want to open web page of a product shown in Fig.4, they can do by clicking picture of the target product, which is embedded with URL link of the product's web page.

Issue 3: How do We Improve Precision of Keyword Search?

Basically, keyword search of shopping websites checks whether product description contains the keywords inputted by the users. Sometimes the search result contains unexpected products. For example, the result of product search by keywords "USB memory 128GB" may contain the result that includes *an adapter* which can be used for USB memory 128GB. To solve this problem, xSearchPro allows the users to define additional search condition (s.a. detailed product category, price, etc. For example, if a user specifies "USB memory" as the target product category, the search result will not contain any adapter.

Seq. No.	Product Picture	Product Code	Product Description	Average Review Score	Number of Reviews	Price including tax	Shipping Cost to Tokyo	Points to be earned	Actual Price	Stock Availability	Expected Shipping Date	Shop Name	Shopping Site
取得番号	商品画像	商品コード	商品名	レビュー評価	レビュー件数	税込価格	東京都への送料	獲得予定ポイント	実質価格	在庫有無	発送目安	ショップ名	出店サイト
R_0001		未取得	送料無料 USBフラッシュメモリ 128GB アルミボディ USB2.0メモリ USBメモリ usb メモリ usbメモリー フラッシュメモリー 小型 高速 大容量 コンパクト シンプル コンパクトセット 2.0 おすすめ	3.75	4件	¥1,350	未取得	13pt (1%)	未計算	在庫あり	未取得	クラッチバッグ elevenone	楽天市場
R_0002		未取得	【サンディスク SanDisk 海外パッケージ】サンディスク USBメモリ 128GB SDCZ50-128G-B35 USB2.0対応	4.50	2件	¥1,384	未取得	13pt (1%)	未計算	在庫あり	未取得	あきばお～楽天市場支店	楽天市場
			SDCZ50-128G-B35【ネコポス便配送制限8個まで】並行										

Fig. 4. An example of a screen shot of product search result in an excel worksheet.

III. System Design of xSearchPro

As we have already explained in the first section, the man-hours of developing xSearchPro should be as small as possible. Furthermore, the users should be able to quickly get used to using xSearchPro. Therefore, we decided to employ Excel macro for developing xSearchPro. Excel macro provides internal functions and many APIs that enable us developing xSearchPro in very short time. Furthermore, the users can use sort and filter function that they are familiar to arrange the search result into the pattern they want. Therefore, we need not to develop this facility.

A. The Input and Output of xSearchPro

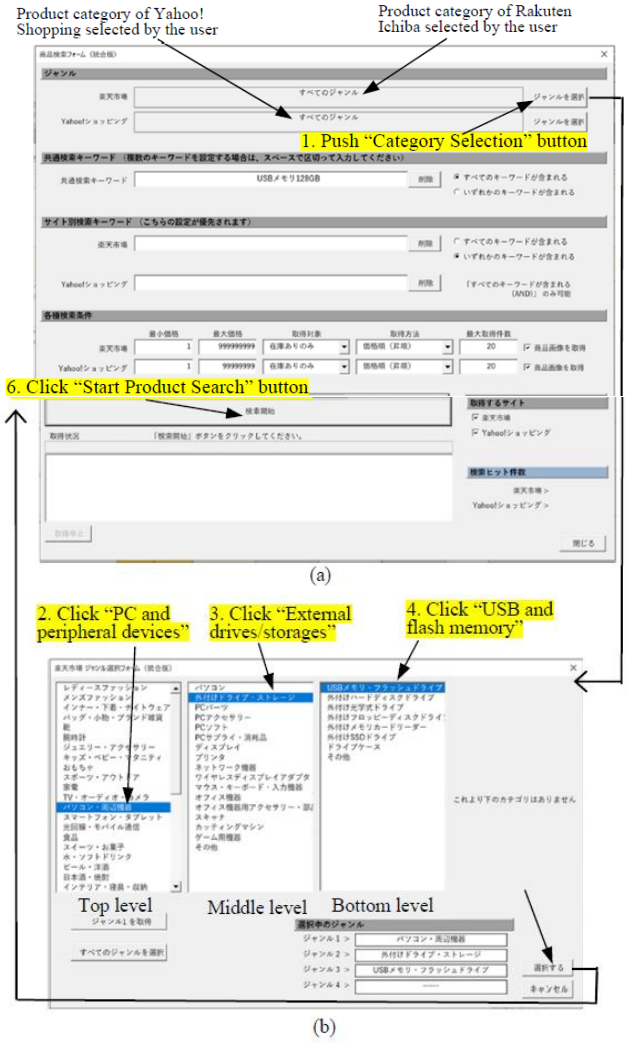

Fig. 5. (a) An example of a screen shot of "Search form" and (b) an example of a screen shot of "Product category selection form".

Since we have already explained the output of xSearchPro in the previous section, we will describe the input of xSearchPro. To improve precision of keyword search, we have designed Search Condition Definition Form (Search Form, for short) shown in Fig. 5(a) and Product Category Selection Form (see Fig. 5(b)) as input parts of xSearchPro. By Search Form, the users can instruct web servers of the two shopping sites to focus scope of product search. The users can restrict number of search result items to reduce time that they must wait for the result. Moreover, the users can use Search Form for defining how search result should be sorted and displayed.

B. The Operation Flow of xSearchPro

Fig. 6 shows the following operation flow of xSearchPro.

(1) A user inputs search condition at the search form provided by xSearchPro and presses the "Start Product Search" button.

(2) xSearchPro sends search requests to Rakuten and Yahoo!. The details of APIs have already described in the previous section.

(3) xSearchPro obtains search result from the APIs. The product information of Rakuten Ichiba is described in XML format while that of Yahoo! Shopping can be described in both XML and JSON formats.

(4) xSearchPro analyzes the XML data of Rakuten Ichiba and the JSON data of Yahoo! Shopping, and stores the product information in an Excel worksheet.

(5) Rakuten Ichiba's API does not support retrieving item_number (product number) like Yahoo Shopping's API does. Both Yahoo! Shopping's API and Rakuten Ichiba's API do not support retrieving values of *shipping_costs* and *estimated_shipping date*. From now on, we will refer to these data as *additional product data*. The user selects the products which they want to obtain additional data.

(6) xSearchPro retrieves the additional product data by analyzing the web page containing the additional data. xSearchPro extracts the target data from HTML source of the product web pages and output it to cells of the worksheet.

(7) The user browses and compares the product information collected by xSearchPro. She can use table data sorting function and filter functions of Excel to find out the desired product information efficiently.

(8) The user can add the product information that she is interested to the favorite worksheet so that she compares and examines the product information later.

(9) The user decides which product should be purchased after comparative investigation.

(10) When the user clicks the product picture to which URL link of the product web page is embedded, xSearchPro instructs web browser to open the actual product web page.

(11) The user performs the purchase procedure and confirm the order.

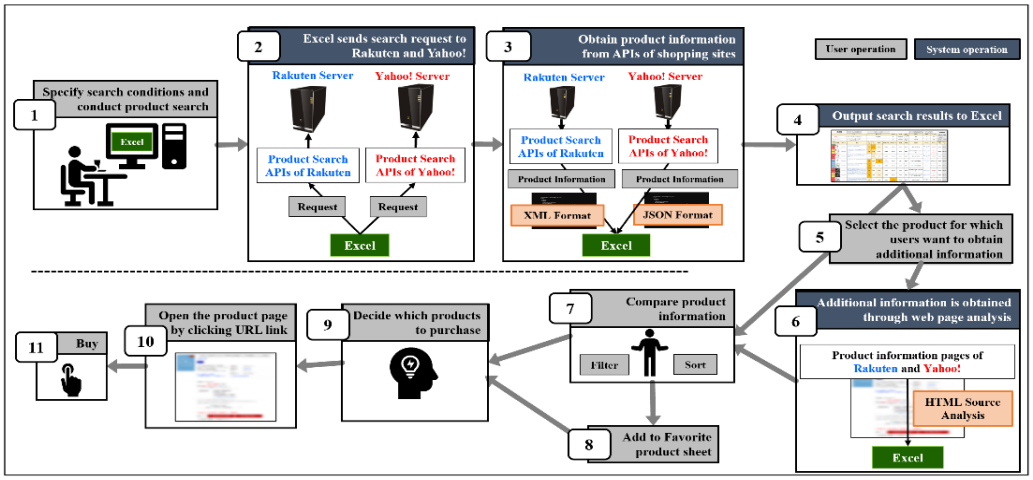

Fig. 6. The operation flowchart of xSearchPro.

C. The Development and Execution Environment

The development and execution environment for this system are described below.

- OS: Microsoft Windows 10 Professional 64-bit
- CPU: Intel Core i7-7500U CPU @ 2.70GHz 2.90GHz
- Main Memory: 8.00GB
- Software: Microsoft Excel for Microsoft 365 [12]
- Program Platform: Excel VBA [13]

IV. Retrieving Additional Product Data by HTML Analysis

In this section, we present a method that we develop to retrieve additional product data by analyzing the web pages of Rakuten Ichiba and Yahoo! Shopping. We implemented this method as an optional function of xSearchPro. Basically, the additional product data of a product can be retrieved from the actual web page of that product. xSearchPro instructs Internet Explorer (IE, for short), which is a browser installed with Microsoft Windows, to open the product web pages. The URL of a product web page can be obtained from Rakuten Ichiba's API and Yahoo Shopping's API. Due to the limitation of paper pages, we will concentrate on presenting a method for retrieving additional product data from product web pages of Rakuten Ichiba. However, the main concept of this method can also be applied for retrieving additional product data of Yahoo Shopping.

After xSearchPro instructed IE to open a product web page, xSearchPro reads HTML source of that web page. In the HTML source, there are elements (in other word, HTML tags) containing an attribute called *class*. For example, Fig. 7(a) depicts values of *item number* and *estimated shipping date* of a product of Rakuten Ichiba. By analyzing HTML source shown in Fig. 7(b), the value of *item number* is defined by the *span* element whose *class* attribute value is set to "*item number*". Therefore, to obtain the value of *item number*, xSearchPro must find for this element by text search and extracts the value of this element.

Based on our investigation, we conclude that *estimated shipping date* is defined by *<td><font size="3" color="#FF0000">* element. This element is located between ** and *個数 *. Therefore, xSearchPro obtains the value of *estimated shipping date* by searching the *font* element that is satisfied by the above condition, and extract value of that element.

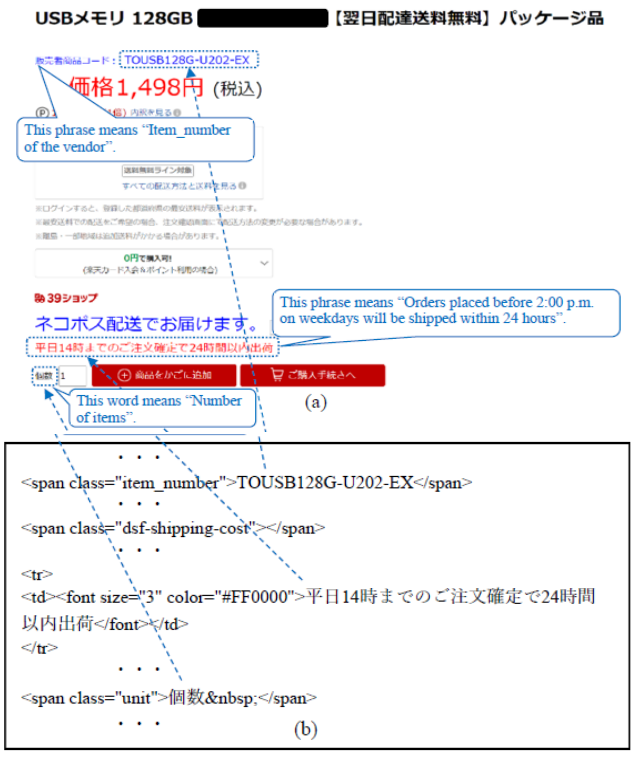

Fig. 7. (a) An example of a part of web page containing *item_number* and *estimated_shipping_date* of USB memory 128GB of Rakuten Ichiba and (b) a part of HTML source of (a).

V. Search Time Evaluation

Two types of experiments have been conducted to measure the process time that xSearchPro spends on searching product from both websites and outputting search result into Excel worksheet.

Experiment 1: Measuring process time when searching by using APIs only.

As shown in Table I, (A) System column depicts the process times of the system for five different product types. For each product type, we measured system's process time in case of collecting information of 100 product items from Rakuten Ichiba and Yahoo! Shopping, respectively. (B) Web column depicts the time required for performing product search at websites and browsing the same search results without recording any product information. By using xSearchPro, the process time of (B) was reduced by about 73%. However, if we add the time that a user must spend on transferring product information from web page to Excel worksheet into the process time (B), xSearchPro is expected to reduce the process time (B) by 99%.

TABLE I. THE PROCESS TIME OF EXPERIMENT 1

Product Type	Process time (seconds)		Time Reduction Ratio = (B-A)÷A
	(A) System	(B) *1 Web	
SSD	15	48	69%
Microwave Oven	16	46	65%
Office chair	14	47	70%
Soy sauce	8	49	84%
Frying pan	10	48	79%
		Average	73%

TABLE II. THE PROCESS TIME OF EXPERIMENT 2

Product Type	Product Name	Number of investigated items		Process time (seconds)		Time Reduction Ratio = (B-A)÷A
		Rakuten Ichiba	Yahoo! Shoppin	(A) System	(B) *1 Web	
SSD	CT1000P1SSD8JP	10	10	156	276	43%
Microwave Oven	M0-F2402	10	10	116	257	55%
Office chair	150-SNCM001	4	2	76	79	4%
Soy sauce	特選丸大豆しょうゆ750ml	10	10	135	243	44%
Frying pan	CF-28B-WBU	6	10	96	219	56%
					Average	41%

Note: 1. The process time (B) does not include the time that a user spends on transferring product information from web page to Excel worksheet.

Experiment 2: Measuring process time when searching by using both APIs and web page analysis

We conducted product search on five different products of Rakuten Ichiba and Yahoo! Shopping. The result of each product search was sorted by descending order of the real prices (computed by *product price + shipping cost – points to be earned*). In this experiment, product information of the top 10 search results of each site is collected. If there are less than 10 search results, all the product information is collected. As shown in Table 2, xSearchPro can reduce the process time of (B) about 41%. However, if we add the time that a user must spend on transferring product information from web page to Excel worksheet into the process time (B), xSearchPro is expected to reduce the process time (B) by 77%. Since web page analysis spends time on reading HTML source, we recommend that users first perform product search by using API to obtain product information without additional information (*item number*, *shipping costs* and *estimated shipping date*). Next the users choose the products that they want to know additional information and use web page analysis function to obtain the additional product data.

VI. RELATED WORK

Typically, product information (e.g., product entity specifications) of shopping websites is structured and stored in relational databases. To fill the gap between the free-form keyword queries and the structured product entities, Duan *et al.* [14], [15] proposed a probabilistic retrieval model. The proposed method mines and analyzes the product search log data to solve the semantic mismatch between queries and structured product entities. Gysel *et al.* [16] introduced a latent semantic entity model to learn the distributed representations of words and entities (i.e., products and queries). However, xSearchPro differentiates itself from the above work by improving precision of keyword queries with hierarchical product category and other metadata.

Huang *et al.* [17] proposed an improved product review search system that facilitates the user to find product reviews on general web search engine. However, finding product reviews which are outside shopping websites is not our research scope.

Ai *et al.* [18] proposed a hierarchical embedding model for personalized product search. The model is a latent space retrieval model which projects queries, users and items into a semantic space and conducts product retrieval according to the semantic similarity between items and the composition of query and user models. However, the drawback of personalized product search is that it limits the users' ability to become exposed to product information that would be relevant to the user's search query. Since some of the product information differs from the user's interests and history, the product information is not displayed to the users. However, xSearchPro is a search tool that releases the users from side effect of personalized product search of shopping sites.

VII. CONCLUSION AND FUTURE WORK

We have presented xSearchPro, a spreadsheet-based search tool that allows cross-searching products of Japan's shopping websites. xSearchPro has the following four outstanding features. Firstly, xSearchPro outputs product information search result including product pictures to a worksheet. The users can use sort and filter functions of spreadsheet software to arrange the search result into the pattern that they can compare the products efficiently. Secondly, xSearchPro has shown good response time in case of searching product information by APIs. Thirdly, xSearchPro is a search tool that releases the users from side effect of personalized product search of shopping sites. Lastly, xSearchPro provides a facility that allows the users to add the interested product information into the favorite worksheet.

This work leaves the following two issues for future development. The first is expansion of xSearchPro to support product searches on other shopping websites such Amazon[6] and Google shopping[7] which provide product search APIs. The second is adding a new function that identifies the same products obtained from shopping websites and displays them together in a continuous manner. If this function becomes possible, the efficiency of product comparison for users will be dramatically improved.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Yamato conducted this research under the guidance of Somchai. Somchai wrote this paper based on a part of

[6] https://www.amazon.co.jp/

[7] https://www.google.co.jp/shopping?hl=ja

Yamato's thesis research (Japanese version), adding the new "Introduction Section", "System Design of xSearchPro Section" and "Related Work Section". All authors had approved the final version.

REFERENCES

[1] Sumitomo Mitsui Card Co., Ltd. (2020). "More than 80% of respondents want to use online services in the future!" [Online]. Available: https://www.smbc-card.com/cashless/knowledge/online_service.jsp?dk=ad_nws_083_91258

[2] R. Baeza-Yates, A. Cuzzocrea, D. Crea, and G. L. Bianco, "An effective and efficient algorithm for ranking web documents via genetic programming," in *Proc. the 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19)*, 2019, pp. 1065-1072.

[3] L. Wu, D. Hu, L. Hong, and H. Liu, "Turning clicks into purchases: Revenue optimization for product search in e-commerce," in *Proc. the 41st Int. Conf. on Research & Development in Information Retrieval (SIGIR '18)*, 2018, pp. 365-374.

[4] B. Li, A. Ghose, and P. G. Ipeirotis, "Towards a theory model for product search," in *Proc. the 20th Int. Conf. on World Wide Web (WWW)*, 2011, pp. 327-336.

[5] B. Long, J. Bian, A. Dong, and Yi Chang, "Enhancing product search by best-selling prediction in e-commerce," in *Proc. the 21st Int. Conf. on Information and Knowledge Management (CIKM)*, 2012, pp. 2479-2482.

[6] Rakuten, Inc, "Rakuten Product Search API (version: 2017-07-06)", [Online]. Available: https://webservice.rakuten.co.jp/api/ichibaitemsearch/

[7] Rakuten, Inc, "Rakuten Product Category Search API (version: 2014-02-22)". [Online]. Available: https://webservice.rakuten.co.jp/api/ichibagenresearch/

[8] Yahoo Japan Corporation, "Product Search API (v3)", [Online]. Available: https://developer.yahoo.co.jp/webapi/shopping/shopping/v3/itemsearch.html

[9] Yahoo Japan Corporation, "Product Category ID Acquisition (v1)", [Online]. Available: https://developer.yahoo.co.jp/webapi/shopping/shopping/v1/categorysearch.html

[10] W3C, Extensible Markup Language (XML) 1.0 (Fifth Edition), W3C Recommendation 26 November 2008. [Online]. Available: https://www.w3.org/TR/xml/

[11] Ecma International, The JSON data interchange syntax, 2nd edition, December 2017. [Online]. Available: https://www.ecma-international.org/publications -and-standards/standards/ecma-404/

[12] A. Kinser, K. Jacobson, E. Kinser, B. Moriarity, and J. Nightingale, *Your Office: Microsoft Office 365, Excel 2019 Comprehensive 1st Edition*, Pearson, June 1, 2021.

[13] J. Korol, *Microsoft Excel 2019 Programming by Example with VBA, XML, and ASP*, Mercury Learning & Information, July 22, 2019.

[14] H. Duan, C. Zhai, J. Cheng, and A. Gattani, "A probabilistic mixture model for mining and analyzing product search log," in *Proc. Int. Conf. on Information and Knowledge Management (CIKM)*, 2013, pp. 2179-2188.

[15] H. Duan, C. Zhai, J. Cheng, and A. Gattani, "Supporting keyword search in product database: A probabilistic approach," in *Proc. of Int. Conf. on Very Large Database (VLDB)*, 2013, pp. 1786-1797.

[16] C. V. Gysel, M. de Rijke, and E. Kanoulas, "Learning latent vector spaces for product search," in *Proc. of Int. Conf. on Information and Knowledge Management (CIKM)*, 2016, pp. 165-174.

[17] S. Huang, D. Shen, W. Feng, C. Baudin, and Y. Zhang, "Improving product review search experiences on general search engines," in *Proc. of the 11th Int. Conf. on Electronic Commerce (ICEC '09)*, New York, USA, 2009, pp. 107-116.

[18] Q. Ai, Y. Zhang, K. Bi, X. Chen, and W. B. Croft, "Learning a hierarchical embedding model for personalized product search," in *Proc. the 40th Int. Conf. on Research and Development in Information Retrieval (SIGIR '17)*, 2017, pp. 645-654.

Somchai Chatvichienchai received the BS, MS degrees in computer engineering from Chulalongkorn University in 1977, Kyushu University in 1989, respectively, and the Ph.D degree in informatics from Kyoto University in 2004. He joined the Department of Info-Media at Siebold University of Nagasaki in 2004. He has become a full-time professor of Dept. of Information and Media Studies since 2009. He has also become a full-time professor of Dept. of Information Security since April 2016. His research interests include database theory and systems, XML, access control, and information retrieval. Dr. Somchai is a member of the ACM, IAENG, and the Database Society of Japan (DBSJ).

Itakura Yamato received the BS degrees in information security science from University of Nagasaki in 2021. His research interests include Web Technology, XML, and information retrieval. He is working as a freelance programmer.