# Design and Implementation of Event Information Summarization System


Chenyi Liao
Graduate School Engineering
Nagoya University
Nagoya Japan
liao@ucl.nuee.nagoya-u.ac.jp

Katsuhiko Kaji
Graduate School Engineering
Nagoya University
Nagoya Japan
kaji@nuee.nagoya-u.ac.jp

Kei Hiroi
Graduate School Engineering
Nagoya University
Nagoya Japan
k.hiroi@ucl.nuee.nagoya-u.ac.jp

Nobuo Kawaguchi
Graduate School Engineering
Nagoya University
Nagoya Japan
kawaguti@nagoya-u.jp



***Abstract*—In this research, we have designed and implemented an Event Information Summarization System (EISS) for collecting Event Info as a web-service. EISS collects mass event data from several non-uniform event website APIs and data sources. The Collected event data is visualized by some user-friendly user interfaces for consumer. EISS can summarize the event data in locational info and temporal info automatically and visualizes them to consumer on online maps. The Event Info is not showed to the consumer by a single list any longer. The consumer will experience the Event Info that is shown by locational online maps. Consumers also can set the query conditions or categories of events to filter out the events info that they need. We also designed and implemented a machine-learning algorithm to estimate the categories of event. EISS results in F1-Score to 0.47 by simple feature. We mentioned that some features are strong and positive correlate with categories expressly.**



***Keywords*—*Event Information Summarization System; Knowledge Processing; Locational Information; Ubiquitous Computing***


## I. Introduction

Recently, the infomation of event is heavily massed on the Internet. There are many websites supporting online services about publishing, advertising, sharing and managing the Event Info. Some of them also support APIs as structured data such as XML or JSON to benefit the secondary developers who want to expand function, to process big data, or to share the Event Info with them.

There are lot of systems for information summarization. Those systems extract knowledge from huge datasets. In the research by S.Ichien and their team[1], they are collecting and summarizing POI(Point of Interesting) data from several data resources. And they advocate publishing those data by Linked Open Data. In the research by W.Chen and their team[2], they have developed a system for processing dish-names. Their system extracts dish-names from online data resources and recommends the dish for users. There are lot of news information collection systems[3]. The POI system and the dish-name system are focused on location information but time information. The news systems that approximating EISS focus on both location and time information but tending to past events.

In this research, the event means a gathering of people who have been invited by a host for purposes of socializing, conversation, or recreation. Different from news, the event presented in form of an event planning is future tense oriented rather than past tense. The event planning includes planning a festival, ceremony, competition, party, concert, or converting with bright categories. In the same way, the event depends on place and time. Therefore it is sensitive to place info and temporal info.

Fig.1 shows an example of event data, which is published by Facebook[4]. This code of event data is complied as a JSON document including a: unique URL, b: title, c: the description, d: place info, e: temporal info and so on. We have analyzed several APIs of event websites. As mentioned above, almost all of them include 5 data fields. However there are some small differences between each APIs, such as the data field name. Therefore, it is necessary to propose a uniform data structure standard being in favor of sharing the data.

Compared with traditional print media, as print advertising and poster, amount of Event Info has been surged on the Internet. However, it is difficult to support consumers with necessary knowledge extracting from mass Event Info accurately. Because of this, Knowledge Processing Technology, which assists consumers to find out and summarize available knowledge from mass data, has been more and more important. For example, the locational info about where the event will happen or the temporal info about when the event will happen is one of the necessary knowledge for consumer. For this reason, we have designed and implemented an EISS[1] for collecting Event Info as a web-service. The system collects mass event data from several non-uniform event website APIs and, summarize the event data as categories and visualizes

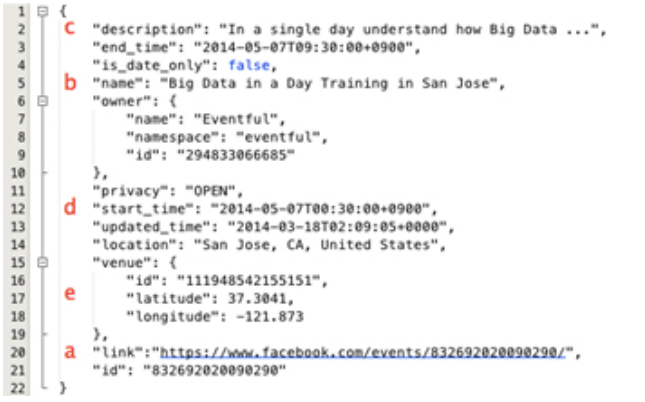


Fig. 1. Example of Event Data from Facebook Event APIs[4]

[1]http://event.locky.jp/

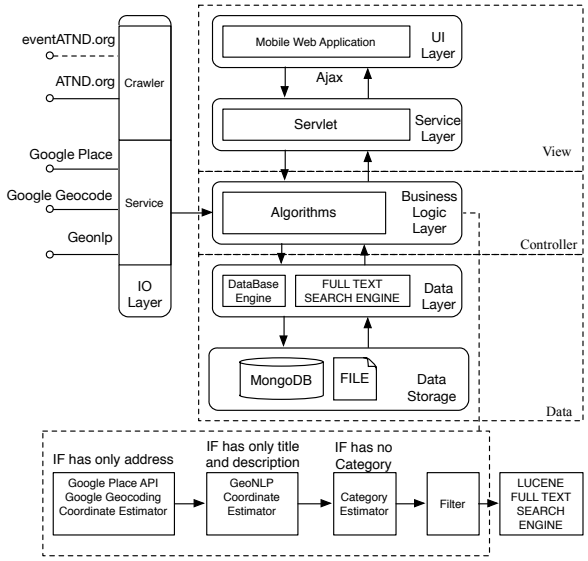


Fig. 2. The Strucure of System and Flow of Event Data Capturing

them to consumer in user-friendly UI.

We describe the design framework of this system in section II, about program flows and data flows – how data flow from online data sources and event websites APIs to UIs through storage pools. In section III, we propose a machine-learning algorithm which is used to estimate the categories from non-classified events. In the section IV, we focus on conclusions and a further study.

## II. Development of Event Information Summarization System

In this section, we describe the design and implement of EISS. First, we explain that why we should built a system that based on ubiquitous computing. Second, we introduce how the crawler captures event data from event website APIs. Third, we explain how to format data structure and store them into our database. Fourth, We introduce our UI design by a mobile divice demo.

With the development of mobile Internet and the popularity of mobile device, the ubiquitous computing is increasing rapidly. Most of online applications are required to support anytime/anywhere/anymedia paradigm in ubiquitous computing[5]. That makes consumers depend on mobile Internet in their daily lives deeply. And the mobile Internet literally altered the life style of consumer. Instead of desktop computing, the ubiquitous computing has an advantage in location information with mobility. By means of location information, ubiquitous computing has given a new way to collect data and feedbacks from consumer[6]. For this reason, we try to design the system that based on both desktop computing and mobile computing as a mobile web application[7].

As shown in the Fig.2, We use MVC (Model View Controller) structure to keep independence between each module. The service layer supports UI Layer to be a web application. It can also support a client application in future. The business logic layer carries a lot of algorithms. IO layer is used to connect with other services and share data. Data layer is used to store data on database or file.

The process of data collection is showed in the bottom sub figure processing from left to right. Each 4 hours, the crawler snatches the event data increment from the each event websites automatically. In this process, we do not get the whole event data from event websites however the data fields that we need. The other data can be obtained by linking data source. We borrowed rules from the RSS standard[8] to constraint data stucture of event data. In this process, if all the data fields title, link, description and date time exist, the event data are seen as available. The event data entities are identified uniquely as an URL. We store data by using Mongo DB[9], which is a NO-SQL DBMS. As a result, the event data are stored in Database as structured JSON documents. The document is identified not only by id of Mongo DB, however also by URL.

Normally, the coordinate data also can be snatched from event website APIs as longitude and latitude. However, not all of situations are ideal. For instance, some events have only address info without coordinate. We estimate the coordinate for address info by Google Place API[10] and Google Geocoding API[11]. Even, some events have no address info. We estimate the coordinate by GeoNLP. The GeoNLP[12] is an online service, which can obtain coordinate from Japanese natural language text, developed by National Institute of Informatics. However, we observed that sometimes it is impossible to estimate coordinate for some events because they have no particular coordinate such as an online event, a television program or error Event Info. We ignore them and they are showed as a special list at our UI.

Categories are very important for classification of event. Parts of the event websites provide categories of event by their APIs. However, some APIs of event website do not provide categories. Therefore we had to develop a machine-learning algorithm to classify them. The algorithm will be explained in section III in detail. The filter module judges whether the event data can be published or not by checking data integrity. In this system, if an event data could be published, the data field title, link, description and date time must exist.

Then, we use those available data to construct the index of full text search engine by Apache Lucene[13]. In this system, the UI is refused to access database directly. When UI request to query the data from database, it must go through full text search engine as a proxy.

The UI design shown in Fig.3 is the homepage of this system, which is displayed on device of Apple iPhone iOS Safari. It is an adaptive layout for PC, Android device, tablet PC and other smart devices. In this page, events are showed on the Google Maps API[14] as markers. The adjacent markers are gathered as a big round icon at a zoom to keep the screen clear.

Fig.4 shows the detailed event markers in near zoom. A red special double-marker icon displays the events that are organized at the same place. User can extend the double-marker icon by touching it. The info window is displayed by touching detail icon. In there, the info window shows the title, date time, address and first 200 words of description. User can link to the page of event by touching the link button.

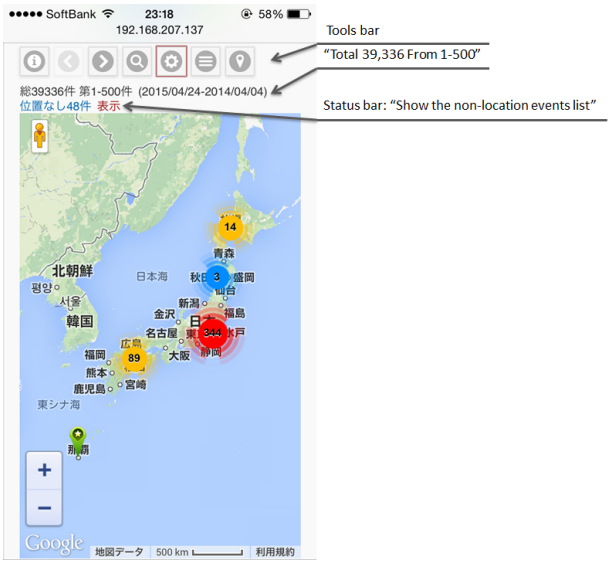


Fig. 3. Example of Homepage


Fig. 4. Example of Event Markers and Info Window

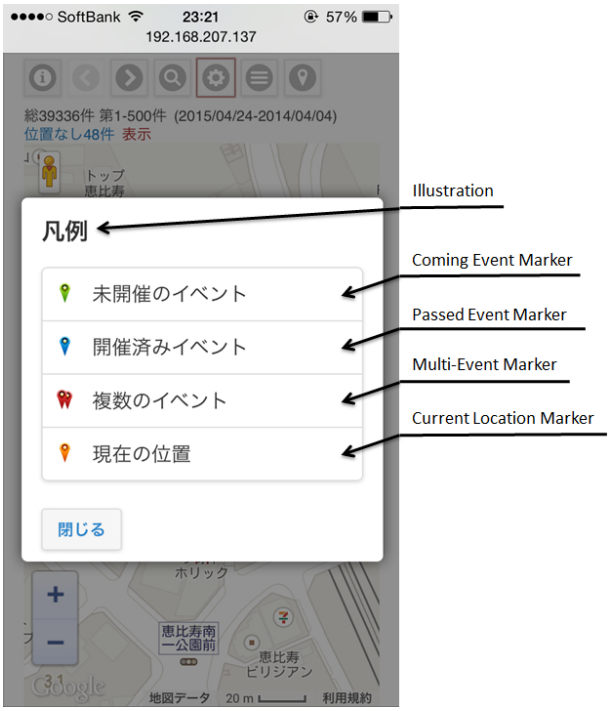


Fig. 5. The Illustrations of Markers


Fig. 6. Example of Events List Ranking by Date Time

There are 4 kinds of marker icons used to mark the events location in this system shown in the Fig.5. The passed events are marked by blue markers, and the coming events are marked but green markers. The red double-marker denotes the events that are organized in the same place. The orange marker icon assigns the location of user currently.

In the section I, we mentioned about two sensitive features for event – temporal info and locational info. Therefore we designed the events list ordered by time shown in the Fig.6.

And another sensitive feature of event is locational info. Consumers always decide to participate in a event by the distance from their current position. We also designed the events list ordered by distance to rank the events that are far from current position shown in the Fig.7.

Both date time list shown in Fig.7 and distance list shown in Fig.7 can be targeted on the maps and the info window is


Fig. 7. Example of Events List Ranking by Distance

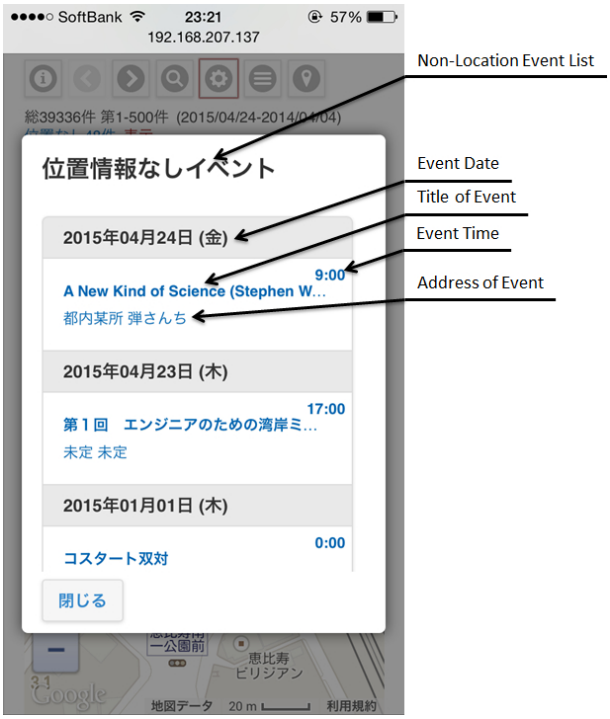


Fig. 8. Example of Non-location Events List

opened when the element of list is touched. We have already mentioned that some events without particular coordinate. To solve this, we design a non-location list to show those events.

As shown in the Fig.8, the non-location events are showed by list. In this case, touching the list element does not get out the event marker on maps. In the meanwhile, the event is redirected to the original website directly by Brower. We observed some of the non-location events with low credibility.

TABLE I. CATEGORIES OF EVENTS

| Categories | English | Categories | English |
|---|---|---|---|
| グルメ | Gourmet | スクール | School |
| アート | Art | お祭り | Festival |
| 展示会 | Exhibition | ショッピング | Shopping |
| ファッション | Fashion | エンタメ | Entertainment |
| 映画 | Movie | ボランティア | Volunteer |
| IT | IT | ビジネス | Business |
| 交流会 | Exchange Meeting | スポーツ | Sports |
| ビューティ | Beauty | くらし | Living |
| 演劇・演芸 | Theater and Entertainment | 音楽 | Music |

Separating the non-location event from home page is conducive to enhance location events ' credibility.

We use SPA (Single-page Application)[15] to optimize the user experience. In this system, either all-necessary code is loaded with a single page including HTML, JavaScript and CSS. The data communications between client and server is supported by AJAX to increase the speed of response.

## III. THE CATEGORIES ESTIMATION ALGORITHM

In this section, we will introduce the background of why we should estimate the categories for Event Info at first. Second, we explain the program flows of training algorithm and estimating algorithm. Third, we discuss the results of experiments.

The category is very important for Event Info. It benefit to consumer to find out and filter the event data easily. The event service website "eventATND", which is one of data resources for us, was providing events info with categories until March 31th 2014 when the service has been terminated. The new service "ATND[16]" did not provide categories any longer. Before the "eventAtnd" service terminated, we had collected 16,063 event data from "eventAtnd" with categories. Therefore, by analyzing events with categories, it is possible to design a machine-learning algorithm to estimate categories of non-classified events.

We found that there are some classification confusions in the old service. Therefore we have defined a closed set of categories instead of filled in by users. In other words, the training sample has noise. During the service "eventAtnd", there were main 18 kinds of category for events had been defined as shown in the TABLE I. Because some events have been taged by special categories such as "その他 (others)", there are 12,988 available event samples left in "eventATND" service.

We propose categories estimation algorithm based on CYK[19] algorithm. The categories estimation algorithm is the analogy of TF-IDF algorithm[17][18], which extracts keywords that appear frequently in a document, however that do not appear frequently in other documents. However, the original TF-IDF algorithm is not fit for categories estimation. TF-IDF algorithm cannot solve the problem of polysemy. As shown in the Fig.9, different from the TF-IDF algorithm, the categories estimation algorithm needs to analyze correlativity between keywords of document and its categories. For example, if TF-IDF extracts a keyword "apple" from the title of Event Info, it may be either an "IT event" or an "agriculture event". In this case, TF-IDF brings ambiguities. However if it appears together with keyword "development", it probably is

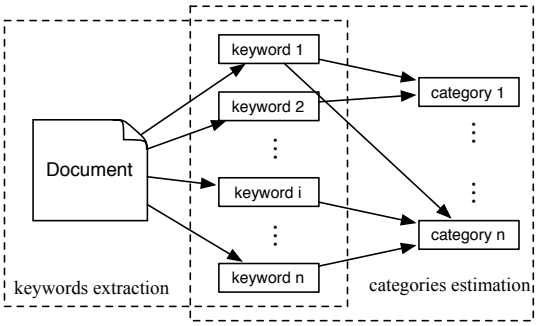


Fig. 9. Keyword Extraction and Categories Estimation

TABLE II. TRAINING ALGORITHM

| | Algorithm 1 Training Algorithm |
|---|---|
| 1 | begin initialize $V_{[len][len]}$ as a string array |
| 2 | i $\leftarrow$ 0 |
| 3 | do $i \leftarrow i+1$ |
| 4 | $V_{i1} \leftarrow \{A \mid A \leftarrow x_i\}$ |
| 5 | until $i = n$ |
| 6 | $i \leftarrow 0$ |
| 7 | $j \leftarrow 0$ |
| 8 | $k \leftarrow 0$ |
| 9 | $V_{ji} \leftarrow V_{jk}$ connects with $V_{[k+j+1][i-k-1]}$ |
| 10 | Store($V_{ji}$, categories++) |
| 11 | until $k<i$ |
| 12 | until $j<len-i$ |
| 13 | until $i<len$ |

not an “agriculture event” however an “IT event”. Therefore we use CYK algorithm to disambiguate polysemy.

Generally, the proper noun as a keyword is more correlative with categories. Significantly different from English, the Japanese language is a non-segmented language. For this reason, Japanese text string has to be transformed into a word tag string at first. We used MeCab[20], which is a Japanese part-of-speech and morphological analyzer system, to segment Japanese sentence. The MeCab labels 4 part of speech tags of proper noun as “Names”, “Organization”, “Place” and “General proper noun”.

As shown in the TABLE II, at line 4, variable “A” means the sentence of event tittle, and the variable “$x_i$” means a word in “$A$”. we have modified the rewriting rule at line 9 to connect two sub strings. And at the line 10, connected two sub strings are stored with the frequency of categories. As a result, the algorithm has generated a CYK table as shown in Fig.10. In this table, the sub string of every unit is stored with its categories frequency.

$$\begin{aligned} P(C|W) &= \frac{P(W_1 \cdots W_i \cdots W_n|C)P(C)}{P(W_1 \cdots W_i \cdots W_n)} \\ &= \frac{P(C, W_1 \cdots W_i \cdots W_n)}{P(W_1 \cdots W_i \cdots W_n)} \\ &= \frac{N(C \cap W_1 \cdots \cap W_i \cdots \cap W_n)}{N(W_1 \cdots \cap W_i \cdots \cap W_n)} \end{aligned} \quad (1)$$

For estimating algorithm, we calculate the probability of every category by Bayesian estimation. As shown in the Eq. (2), “$C$” means one category of 18 kinds of categories. “W”


Fig. 10. Example of CYK Table


Fig. 11. Example of Probality Caculation

means the sub string in title. the probability of a category is that the mutual appearance times of sub string and the category divided by the times of sub string.

Fig.11 showed a example of probability calculation. For this example, there are 45 events have the keyword “web” in their titles. And there are 44 events belonged to the category “IT”. Therefore the probability $P(\text{IT}|\text{Web})$ is $44/45 = 97.78\%$. And there are 17 events belonged to the category “Business”. Therefore the probability $P(\text{Business}|\text{Web})$ is $17/45 = 37.78\%$.

We also desinged estimating algorithm by an inverse algorithm of CYK algorithm, as shown in TABLE III at the line 10 and the line 11, it calculates the max probability of each categority for event. In the end, we set the dicision boundary of $probalility>0$ to select all the categories.

$$\begin{aligned} P &= \frac{N(E \cap I)}{N(E)} \\ R &= \frac{N(E \cap I)}{N(I)} \\ F_\beta &= \frac{(\beta^2+1)PR}{\beta^2 P + R} = \frac{2PR}{P+R} \qquad (\beta = 1) \end{aligned} \quad (2)$$

Depending on training and estimating algorithm, we designed experiments. We used 90% (11,689) of event data as trainning data and others 10% (1,299) of event data as testing

TABLE III. CATEGORIES ESTIMATION ALGORITHM

| | Algorithm 1 Training Algorithm |
|---|---|
| 1 | begin initialize $V_{[len][len]}$ as a string array |
| 2 | i $\leftarrow 0$ |
| 3 | do $i \leftarrow i+1$ |
| 4 | $V_{i1} \leftarrow \{A \mid A \leftarrow x_i\}$ |
| 5 | until $i = n$ |
| 6 | $i \leftarrow 0$ |
| 7 | $j \leftarrow 0$ |
| 8 | $k \leftarrow 0$ |
| 9 | $V_{ji} \leftarrow V_{jk}$ connects with $V_{[k+j+1][i-k-1]}$ |
| 10 | if $P(V) < (V_{ji}, categories++)$ |
| 10 | then $P(V) \leftarrow (V_{ji}, categories++)$ |
| 11 | until $k < i$ |
| 12 | until $j < len - i$ |
| 13 | until $i < len$ |

TABLE IV. RESULTS OF EXPERIMENTS

| Seq. | Part of Speech | Precision | Recall | F1-Score |
|---|---|---|---|---|
| 1 | N+O+P+G | 90.26% | 32.10% | 0.4736 |
| 2 | O+P+G | 90.63% | 32.02% | 0.4732 |
| 3 | N+O+P | 95.73% | 31.10% | 0.4695 |
| 4 | N+O | 95.72% | 31.02% | 0.4686 |
| 5 | N+O+G | 94.04% | 30.95% | 0.4657 |
| 6 | O+G | 91.65% | 30.41% | 0.4567 |
| 7 | O+P | 92.04% | 30.25% | 0.4553 |
| 8 | O | 96.18% | 29.10% | 0.4468 |
| 9 | N+P+G | 49.02% | 1.92% | 0.037 |
| 10 | N+G | 48.98% | 1.85% | 0.0357 |
| 11 | N+P | 48.89% | 1.69% | 0.0327 |
| 12 | P+G | 45.00% | 1.39% | 0.027 |
| 13 | G | 43.59% | 1.31% | 0.0254 |
| 14 | P | 43.86% | 1.15% | 0.0224 |
| 15 | N | 70.00% | 0.54% | 0.0107 |

N: Names Noun O: Organization Noun P: Place Noun G: General Proper Noun

data. By trainning, the estimating algorithm can estimate categories for 10% testing data. Then we compare the estimated categories and initial categories and calculate the precision, recall, F1-Score as shown in the Eq.(2). The "E" means the set of estimated categories. The "I" means the initial categories, which was filled by user.

The results of experiments are shown in TABLE IV. When the part of speech include "Organization Noun", there is significantly increase at F1-Score, as shown in row from 1 to 8. For this reason, "Organization Noun" is strong and positive correlate with categories expressly. Increasing features can improve the F1-Score. We are trying to increase the features, including proper noun, general noun and verb. We will also try to train system by not only the title of event, however also by the description and other data fields.

## IV. CONCLUSION

In this paper we described the design and implement of EISS. We explained how the system can collect event data from online event sites and how the system visualizes event data to final users. We described how the categories estimation algorithm works.

The system has be installed online in end of April 2014, as a pre-alpha test version. The system has collected more than 40,900 event data in area of Japan from years 2010 to years 2015 with data fields of title, time and describe and so on. The target user is for Japanese mainly. With the limitation of current computational capabilities, the recall of categories estimation algorithm is still lower. We need time to analyze more features of POS and data field and try more decision boundary. Through the optimization algorithm, we will focus to advance the performance of the categories estimation algorithm.

In the other hand, we will collect the data exhaust[21] from user behaviors. We also get the user profiles from SNS by OAuth2.0 authentication. Through analyzing the user behaviors and user profiles, we will design the event recommendation system for user. The recommendation system will assist user to select suitable events. As usual, consumer always selects events with similar theme. Therefore, after analyzing the former consumer behaviors, it is possible to recommend new events for consumer who has the similar behaviors[22].

## ACKNOWLEDGMENT

A part of the research is supported by Microsoft Research Asia.

This project is partly supported by Stragetic Information and Communications R&D Promotion Programs (SCOPE) 132306007 in Japanese Ministry of Internal Affairs and Communications.

## REFERENCES

[1] Shinji Ichien, Katsuhiko Kaji, Nobuo Kawaguchi: Proposal of a Platform Integrating POI Information, ICMU, pp124-128, 2014.

[2] Weichang Chen, Katsuhiko Kaji, Nobuo Kawaguchi: Non-Local Dictionary Based Japanese Dish Names Recognition Using Multi-Feature CRF frame Online Reviews, WIMS, 2014.

[3] Newspapermap: http://newspapermap.com/

[4] Facebook: https://facebook.com/

[5] Wonjae Lee, Hyun-Woo Lee, Min Choi, Jong Hyuk Park, Young-Sik Jeong: Hierarchical Customization Method for Ubiquitous Web Applications, CSA 2013, pp 603-608, 2014.

[6] Shinji Ichien, Katsuhiko Kaji, Nobuo Kawaguchi: Proposal of a Platform Integrating POI Information, ICMU, pp124-128, 2014.

[7] Pendyala, V.S., Shim, S.S.Y.: The Web as the Ubiquitous Computer. Computer 42(9) (2009).

[8] The application / rss+xml Media Type. Notwork Working Group. May 22th 2006

[9] MongoDB: https://www.mongodb.org/

[10] Google Place API: https://developers.google.com/places/

[11] Google Geocoding API: https://developers.google.com/maps/

[12] GeoNLP: https://geonlp.ex.nii.ac.jp/

[13] Apache Lucene: http://lucene.apache.org/

[14] Google Maps API: https://developers.google.com/maps

[15] A Mesbah, A van Deursen: Migrating multi-page web applications to single-page Ajax interfaces, Software Maintenance and Reengineering, 2007. CSMR '07. 11th European Conference on, pp181-190, 2007.

[16] ATND: http://atnd.org/

[17] Matsuo Yutaka, Mitsuru Ishizuka. "Keyword extraction from a single document using word co-occurrence statistical information." International Journal on Artificial Intelligence Tools 13.01 (2004): 157-169.

[18] Liu, Fei, Feifan Liu, and Yang Liu. "A supervised framework for keyword extraction from meeting transcripts." Audio, Speech, and Language Processing, IEEE Transactions on 19.3 (2011): 538-548.

[19] Richard O.Duda, Peter E.Hart, David G.Stork, Pattern Classification (2nd Edition), Wiley-Interscience, pp426-429, 2000.

[20] MeCab: http://mecab.googlecode.com/

[21] Viktor Maver-Schonberger, Kenneth Cukier: Big Data: A Revolution That Will Transform How We Live, Work, and Think. Eamon Dolan/Houghton Mifflin Harcourt, 2013.

[22] Michael R.Solomon: Consumer Behavior Buying, Having, and Being (Eighth Edition). Prentice Hall,pages.8-9.2008.