App.Locky: Users' Context Collecting Platform for Context-Aware Application Recommendation

Katsuhiko Kaji
Graduate School of Engineering, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, Japan
kaji@nuee.nagoyua-u.ac.jp

Motoki Yano
Graduate School of Engineering, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, Japan
yano@ucl.nuee.nagoyua-u.ac.jp

Nobuo Kawaguchi
Graduate School of Engineering, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, Japan
kawaguti@ nagoyua-u. jp

ABSTRACT
Recently, it has become very hard for users to find their desired mobile services because the number of applications and Web services are rapidly increasing. Therefore, it is important to realize context-aware application recommendation. Because it is necessary to collect large learning data to estimate user's context, we propose a platform for collecting users' context and relationship between context and application by providing an application search system that inquires user's current context. We implemented a system named "App.Locky" based on our proposal and conducted experiments by publishing the system on the internet. As a result, we confirmed that collected search logs can be used to estimate user's context and relationship between context and application.

Author Keywords
H.4.0: Information Systems: Information System Applications: General

ACM Classification Keywords
Design

General Terms
Crowdsourcing, tagging, context awareness, application recommendation

INTRODUCTION
Recently, there are many high performance portable terminals such as smart phones. We are able to use various kinds of services anywhere. Actually, applications and services for portable terminals are explosively increasing. However, we don't have effective ways to find available applications and services from large repositories. Therefore, application recommendation according to user's situation is needed. There are several methods of context-aware application recommendation. Generally, the systems need large amount of training data for recommendation.

Our final goal is to realize a context-aware recommendation of smart phone applications. In this paper, we propose an application retrieval system named "App.Locky" that recommends smart phone applications according to user's query of current context. User's context is represented as multiple "context-tags". The tags become the media between the user's context and the applications. Selected tags and context information such as location and time are collected to the server. Collected information is available to detect user's context and suitable applications for each context.

RELATED WORK
There are various researches on how to recommend applications according to user's context [1-3]. To realize context-aware application recommendations, information of a user's current context and the relationship between the context and applications is necessary.

There are two ways of getting a user's context and profile. They are explicit and implicit methods. In explicit method, a user's context is input by the user directly [4]. In implicit method, a user's context is automatically estimated [5].

There are two representative methods of application recommendation. One is a method based on the application's features. Each application is associated with a certain context in advance. When the user's context is acquired, the method calculates the nearest application to the context. The method has a problem on how to associate the context and the application. Several researches solve the problem by using a great amount of application usage history [6]. The other is collaborative filtering. The method supposes that valuable applications of similar users in similar context are suitable for the user. Amazon [7] adopts the method to recommend their commercial products by using a great amount of purchase/web-browsing history.

Consequently, to recommend an application that is suitable for a user's context automatically from various applications, it is necessary for a lot of training data such as user's context and relationships between context and application

UbiCrowd'11, September 18, 2011, Beijing, China.

previously. However, there are few researches and systems to collect such valuable data.

APP.LOCKY

We propose an application retrieval system named "App.Locky" that is for smart phone users. A user can retrieve a smart phone application that is suitable for his/her context via the App.Locky. The system is also able to collect smart phone user's context and relationship between context and application.

We collected 4000 applications information from Apple's AppStore. There are 20 categories in AppStore. We picked up top 100 pay/free applications in each category. We also collected general user's review text for the applications via the web.

Context-Tag

Location and time information are representative examples of a user's context. The information can be acquired from smart phone's sensors. However, the information is not enough to represent the whole of the user's context. The current version of App.Locky collects GPS / WiFi location and clock information.

We adopt a "context-tag" to represent user's context. For example, a context-tag can be "during lecture," "party," "hungry" and "lunch." We can treat user's context by context-tag that smart phone's sensors cannot represent sufficiently. Prepared context-tag list is shown in Figure 1. The context-tag tree is based on Japanese Lexicon [8] and Yahoo! Category [9]. Table 1 shows the number of context-tag in each question class. Of course, the context-tag tree is not perfect to represent every user's context. Therefore, App.Locky enables to input new context-tag.

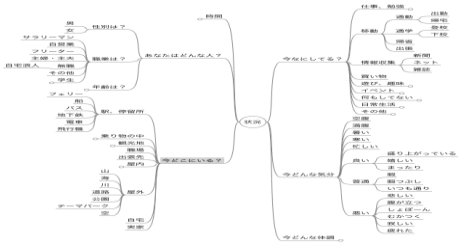

Figure 1: A part of context-tag tree

Question Class	# of context tag
Location	73
Action	129
Mind	20
Health	11
User type	27

Table 1: Context-tag classification and numbers of tags

System Architecture

App.Locky consists of a server and an iPhone/iPod touch application (Fig. 2). The application retrieval sequence is as follows. First, the client application requires query and context-tag information to query controller. Then, the query controller generates queries about a user's context such as "Where are you?" and "What are you doing?", and send the question and the candidate of the answer as context-tag list. Next, the client application shows the questions and tag lists, and the user selects his/her current context by selecting tags. When the selection is finished, the selected context-tag list and location/time information are sent to the server. Next, the search engine retrieves an application's information from the database and sends the result to the client application. The application information database contains application information and the application's review text collected via web. Finally, the client application shows the result. Detailed information can be shown by selecting each application. If the user is interested in the application, the application can be installed. If the user does not like the application, he/she simply presses "don't need" button. The operation is stored in the search log database.

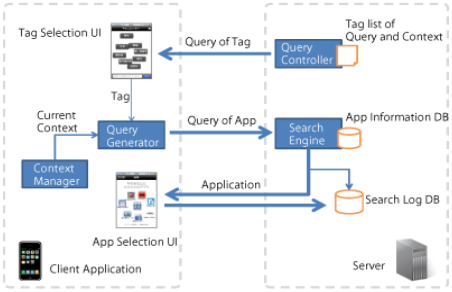

Figure 2: System architecture of App.Locky

Figure 3: Screenshot of selecting current context (left), an example of application recommendation according to current context (right)

User Interface

Figure 3 shows the interface of App.Locky. The upper text is a question of user's context such as "What are you doing?" and "Where are you?" Context-tags to the question class are shown in the center area. The user inputs his/her current context by selecting the tag. As described in 3.1, context-tags are prepared as tree structure. Therefore, if the selected context-tag has child context-tags, the child context-tags are shown in the next phase. Consequently, user needs multiple selections of context-tags. When the user cannot find a suitable tag to represent his current context, the user inputs a suitable context-tag from a text

field as shown on the bottom of Figure 3 left. As the result of application retrieval, each application is shown as an icon (Fig. 3 right).

Algorithm of Application Retrieval

To retrieve a suitable application for the user's current context, the ratio of selected context-tag including the application's review text is considered. The ratio is calculated by using tf-idf. *Score*(a,T) means the score of application by retrieving context-tag list T. Score calculation formulas are as follows.

$$R_a = \{r_i\} \quad (1)$$

$$tfidf(t,a) = tf_{t,R_a} \cdot idf_t \quad (2)$$

$$tf_{t,Ra} = \log \frac{n_{t,Ra}}{\sum_k n_{k,Ra}} \quad (3)$$

$$idf_t = \log \frac{|R|}{|\{r_i : t \in r_i\}|} \quad (4)$$

$$f(t,a) = \begin{cases} 1 & if \quad t \in R_a \\ 0 & if \quad t \notin R_a \end{cases} \quad (5)$$

$$Score(a,T) = \sum_{t \in T} f(t,a) + tfidf(t,a) \quad (6)$$

R_a means review set of application *a*. $n_{t,Ra}$ means the appearance count of context-tag t in R_a. |R| means total review count of all application. $|\{r_i : t \in r_i\}|$ means the review count including tag *t*. When the client application sends a context-tag set T, the server calculates the score for all applications, and returns top 10 applications as retrieval result.

EXPERIMENTS

We released client application of App.Locky to AppStore Japan on December 23, 2010. By 24th January 2011, the total number of downloads was 6557. Table 2 shows an overview of query log.

We verified the statistical nature of collected application's search log. In the experimental period, iPhone users sought applications via App.Locky over 20000 times. At first use, the client application confirms corresponding terminal id and query. The number of users shown in Table 2 is the count of corresponding terminal id and the query. Users watched the detail information of retrieved application 19628 times. Users also got suitable application for the users' current context 2793 times.

Over 250 tags were selected over 10 times. Table 3 shows top 5 most selected tags in each query class and their selected count. As you can see, there are frequently selected tags. Especially, context-tag "home" is selected over 7000 times and the selection ratio is highest. It is not so good to have only a few tags selected many times because our purpose is to collect various kinds of information about user's context. As future work, we should implement some ideas into the App.Locky to be used in various kinds of contexts.

Total # of Query	21382
Query as current context	15130
Query as current context including location	11565
User	4139
Checked detail information of application	19628
Selected "Get the app"	2793
Selected "Not wanted"	1136

Table 2: Overview of query log

Location		Action		Health	
Home	7035	Leisure	3746	Usual	2962
Train	2067	Internet	1159	Sleepy	1782
Office	1358	Nothing	926	Tired	1659
Family home	1099	Homecoming	763	Good	1106
Business trip	313	Sleeping	539	So-so	476
Total	15102	Total	15102	Total	9191
Created	376	Created	698	Created	219

Mind		User type	
Cold	1366	Man	332
Full stomach	688	Business person	131
Hungry	598	30's	104
Busy	438	20's	91
Killing time	293	Woman	75
Total	5478	Total	1309
Created	144	Created	58

Table 3: Top 5 selected tags and their selected count

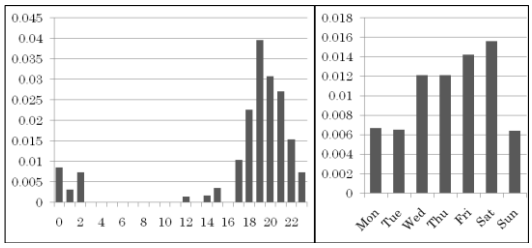

Figure 4: The probability of a tag "Dinner" selected by hour and weekday

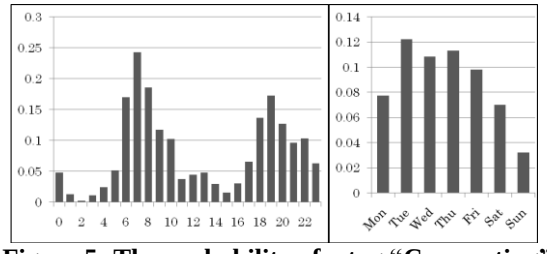

Figure 5: The probability of a tag "Commuting" selected by hour and weekday

Time Dependency of Tags

In this section, we tested time dependency of tags. App.Locky is used frequently at nights of weekends. Figure 4 and Figure 5 shows examples of time dependency of certain tags. As you can see, the probability of these tags by time represents the feature of the context. Like these tags, we found that there are several time dependent context-tags. These tags can be estimated from time information.

Location Dependency of Tags

In this section, we tested location dependency of tags. A characteristic feature is that there are commonly retrieved locations. One of the locations is station. Therefore, we analyzed context-tag selection aspect around station within 300m in Japan. Table 4 shows top 5 tags of each query class that is selected around station. In location class, the tags such as train and subway appeared. These tags are related to station. Moreover, we can find that several tags are related to station in the other question classes. Many users commute by train. Many older people and business people appeared in station. In the context of commuting, we think that many people feel tired so that sleeping is the most frequent action in train. As a result, context-tag related to train is frequently selected around station. Therefore, if the user is around station, we can estimate that the user is in such contexts.

Location	Action	Health	Mind	User type
Mountain	Sleeping	Tired	Hot	40's
Station or Bus stop	Train (hobby)	Usual	Hungry	60's
Subway	Homecoming	Good	Sickness	Business person
Train	Commuting	Stomachache	Usual	30's
Station	Leave school	Sleepy	Full stomach	Man

Table 4: Top 5 tags around stations within 300m

Relationship between Application and Tag

In this section, we tested the relationship between selected context-tags and the applications where the users selected "get the app" tag. Table 5 shows examples of top 5 high co-occurred tags with application. The applications "Tabelog", "NAVITIME" and "Weather News" are the most downloaded applications via App.Locky. As a result, we can say that co-occurred context-tags represent the feature of application to some level.

Consequently, we could collect large-scale search log via App.Locky. By analyzing the collected log, we confirmed the context-tag's time/location dependency, and relationships between tags and applications.

Tabelog	NAVITIME	Weather News
Nearby restaurant search engine	Route navigation application	Weather forecast application
Lunch	Station or bus stop	60's
Cooking	Bicycle	Out door
Japanese food	Gamble	Good
Lack energy	Flight	Soccer
Restaurant	Business trip	Home

Table 5: Top 5 high co-occurred tags with application

CONCLUSION

We proposed an application retrieval system named App.Locky where a user can retrieve a suitable application for his/her current context. Additionally, the system collects a search log of context information and suitable application for each context. We released App.Locky to AppStore and the application has been downloaded by over 6500 users. Additionally, users have retrieved applications via App.Locky over 20000 times. We also analyzed the log and found that there are several time-dependent and location-dependent context-tags. These contexts can be estimated by time and location. Future work is to realize automatic context-aware application recommendation system by using the collected information.

REFERENCES

1. Naganuma, T. and Kurakake S.: Task Knowledge Based Retrieval for Service Relevant to Mobile User's Activity, In Proc. of the 4th Int. Semantic Web Conference (ISWC'05), Y.Gil et al. (Eds.), LNCS 3729, pp.959-973, 2005
2. Zheng, Z., Ma, H., Lyu, R. M., and King, I.: QoS-Aware Web Service Recommendation by Collaborative Filtering, IEEE Transactions on Services Computing, Volume 4 Issue 2, 2011.
3. Setten, V. M., Pokraev, S. and Koolwaaij, J.: Context-Aware Recommendations in the Mobile Tourist Application COMPASS, Adaptive Hypermedia and Adaptive Web-Based Systems, Lecture Notes in Computer Science, Vol.3137, pp.515-548 (2004).
4. Ken, L.: News-weeder: learning to filter netnews, Proceedings of the 12th International Conference on Machine Learning, pp.331-339 (1995).
5. Siewiorek, D., Smailagic, A., Furukawa, J. and et al.: SenSay: A Context-Aware Mobile Phone, Proceedings of the 7th IEEE International Symposium on Wearable Computers, ISWC '03, IEEE Computer Society, p.248(2003).
6. Niklas, J., Hagen, W.S., Christoph, M.M. and et al.: Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations, Proceeding of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion, TSA '09, ACM, pp.57-64 (2009).
7. Linden, G., Smith, B. and York, J.: Amazon.com recommendations: item-to-item collaborative filtering, Internet Computing, IEEE, Vol.7, No.1, pp.76-80 (2003).
8. Shirai, S., Ooyama, Y., Ikehara, S., et al.:Introduction to "Goi-Taikei:A Japanese Lexicon", IPSJ-SIG-IM Vol.98, No.106, pp.47-52 (1998)
9. Yahoo Japan Corporation: Yahoo! Category, available from http://dir.yahoo.co.jp/ (accessed 2011-3-15)