Development of a Recommender System based on Extending Contexts of Content and Personal History

Katsuaki Tanaka
RCAST, the University of Tokyo, Tokyo, Japan
Email: katsuaki@ailab.t.u-tokyo.ac.jp

Koichi Hori
School of Engineering, the University of Tokyo, Tokyo, Japan
Email: hori@computer.org

Masato Yamamoto
NTT DOCOMO, Inc., Tokyo, Japan
Email: yamamotoma@nttdocomo.co.jp

Abstract— **The flood of information on the Internet makes a person who approaches it without some strong intention feel overwhelmed. One way to redress the balance between a person and the flood is a computer-based recommender system, and many Web sites use such systems. These systems on a Web site work for similar items. However, the field of personal activity is not limited to handling one kind of knowledge or one Web site, but also involves off-line activities in the real world. To handle personal off-line activity, LifeLog was proposed as a method to record it, but the main purpose of LifeLog is to record a personal history. The uses of such a history are still being studied.**

We have developed a recommender system that captures personal context from a history of personal online and off-line activities, treats information on Web sites as a large set of context, and discovers and extends the overlaps of personal activities and Web sites, then recommends information located in the Web sites. The aim of the system is to allow a person to enjoy waves of information again.
The system was implemented as part of the *My Life Assist Service* for mobile phones provided by NTT DOCOMO, Inc. as a field experiment from December 2007 to February 2008.

Index Terms—**recommender system, mobile phone, context awareness, LifeLog**

I. INTRODUCTION

There is too much content on the Internet, so people who consult it can lose sight of their own objective without strong intention. A similar thing can also occur on a Web site. To solve this issue, some Web sites have employed recommender systems such as collaborative filtering [1] to help a user browsing the Web site.

Thus, using a recommender system for computer-based content selection and presentation is a way to strike a balance between content generated by other people and a user [2]. Usually, the target of a recommender system is a single Web site.

Such recommenders are sufficient for a person whose activity is limited to a Web site. However, most human activity occurs in the real world. To capture such human activity, the LifeLog was proposed [3]. The main topic of LifeLog research is how to capture activity. The present focus of study is the reuse of captured records.

We developed a context-based recommender system that derives content recommendations from a user's activities as captured by a mobile phone, as part of the *My Life Assist Service*. The service was a field experiment performed from December 2007 to February 2008 by NTT DOCOMO, Inc. and other collaborative partners as a part of the *Information Grand Voyage Project* promoted by the Ministry of Economy, Trade and Industry in Japan. The service was a platform that included specialized services for content providers and consumers. The service for consumers was named the *Preview channel*.

Typically, a recommender system recommends pages from the Web site where the user has been browsing, so the user's needs would be clear, and the purpose of a recommender system is that a user accepts the recommendation and makes an action such as buying something. Therefore, precision enhancement of the recommended content acceptance is the first goal.

On the other hand, the Preview channel was quite different from other services such as searching or recommending content for a user based on the user's location as identified by the user's mobile phone. The genres of recommendation were not given and the recommended content was always displayed on a phone screen, as shown in Figure 2 and was updated only four

This paper is based on “A Recommender System Based on Context Extending of Content and Personal History,” by K. Tanaka, K. Hori, and M. Yamamoto, which appeared in the Proceedings of the International Conference on Information and Communication Systems (ICICS), Amman, Jordan, December 2009.

doi:10.4304/jetwi.2.3.197-203

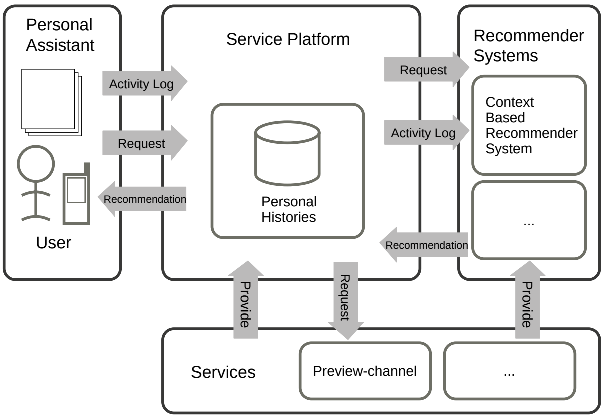


Figure 1. Components of the My Life Assist Service

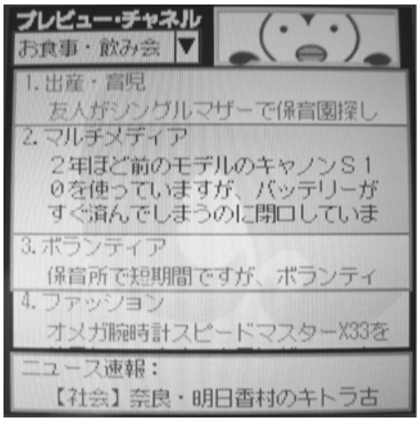


Figure 2. Screenshot of the Preview channel

times a day. Therefore users became bored with precise but unchanging recommendations [4]. The context-based recommender system therefore did not aim for precise recommendations, but recommendations that shifted with the user's interest. We felt that content expresses part of its creator's intentions, and should be located in a context. For example, a promotion for a shop has contexts not only of things that the shop wants to sell but also of the intent in opening the shop, reasons for selling the item, and so on. The context-based recommender system shifted with the user's interests by finding overlaps between the context of a message and the context of the user, and extending them.

II. The My Life Assist Service

A. Components of the My Life Assist Service

The My Life Assist Service consisted of the components shown in Figure 1. The Personal Assistant Application on a mobile phone collected user activities and sent them to the Service Platform. The Recommender System scraped personal history collected from the user's activities, and selected content for recommendations. The Service Platform sent details of requested content to the Recommender System, and sent the answer to the Personal Assistant Application.

The Personal Assistant Application collected locations from the global positioning system (GPS) receiver installed in the mobile phone, content that the user had read on the Preview channel, and texts that the user wrote to other services in the My Life Assist Service, as personal history.

B. Preview Channel

The My Life Assist Service was a platform that included specialized services for content providers and consumers. The Preview channel was a service for consumers designed by NTT DOCOMO, Inc., and the content for the service was provided by *Oshiete! Goo*[1] by NTT Resonant Inc. and *Walkerplus*[2] by Kadokawa Cross Media Co., Ltd.

The Preview channel was provided as an application for NTT DOCOMO's mobile phones. The application (Figure 2) was always displayed on the screen. The application displayed four categories; each category included four brief summaries of content. The user could select a summary and access the relevant content via the phone's Internet access function. The recommender system selected four messages for each of the four categories, for a total of 16 messages.

Because the service always displayed recommended content on the screen, it was considered successful when at least one message attracted the user's attention.

C. Content for Recommendation

Sources of the content were Oshiete! goo and Walkerplus. Oshiete! goo has many Q&A dialogs across many categories; the recommender system used 107 of the categories. Walkerplus consists of *Gourmet Walker*, *Hotel Walker*, and *Wedding Walker*; the recommender system treated each of them as a category. The number of categories that the system used was therefore 110. On the Preview channel, making a recommendation meant selecting four categories from the 110, and choosing items from each of the selected categories.

For example, an item in Walkerplus described the detail of a restaurant. It contained the latitude and longitude of the location. On the other hand, an item in Oshiete! goo is generated when a user of the Oshiete! goo Web site wrote a question and other users answered it. Items did not contain latitudes or longitudes originally, but the service platform added locations to each item that mentioned places or addresses. The recommender system was required to recommend as many items with no locations as with locations.

D. Timings to Update Recommendations

The recommender system was required to update recommendations four times a day. The recommender system can specify the timing of updates. Personal histories are stored in the platform that is updated every three hours.

Recommendations did not change when the personal history did not change because the recommender system

[1] http://oshiete.goo.en.jp/
[2] http://www.walkerplus.com/

Figure 3. Outline of the context-based recommender system

makes recommendations based on personal history. The system should update recommendations at least at three-hourly intervals. There is little flexibility to choose timings to match four updates with three-hour intervals in the daytime. Therefore, the recommender system did not infer timings of updates for each user, but fixed the update times to 8, 11, 15, and 19 hours each day.

E. Design of the Recommender System

The Service platform stored the personal history of each user. The personal history was collected by the Personal Assistant Application and consisted of locations, items that the user had read on the Preview channel, and text items that the user wrote to other services in the My Life Assist Service.

When the My Life Assist Service was designed, it was assumed that most of the personal history would be locations in the real world that were collected automatically every 30 minutes.

We could not make any assumptions about location history, because the history would be quite different for each user, the history items could be grouped for some reason, and we could not find relations within these groups and so on in advance because we had no previous location histories of many persons. Therefore we could not design a user model or create a learning mechanism based on user histories.

We therefore designed a context-based recommender system that did not use generalized statistical models determined from many users' histories, but made personalized models.

III. CONTEXT-BASED RECOMMENDER SYSTEM

The context-based recommender system aimed at nonconverged recommendations to shift with the user's interest.

First, it articulated (see Section III-B) the personal history of activities of the user and the content for the recommendation source. It extracted contexts from them and generated a graph for each personal history and Web page. Then, it found overlaps between the graphs, extended the overlap together with the context of a Web page and recommended content from this extended area.

Several methods have been proposed to provide variation to recommender systems based on attributes of recommended targets or relations in described texts [5]. The feature of the context-based recommender system was that it extended the recommendation area together with the context of the content by inserting two graphs that represented the contexts of the personal history and the content (Figure 3).

A. Inputs to the System

The recommender system used the personal history of activities of each user and the content for recommendation.

1) Personal History: The personal history of each user consisted of two parts: on the Web and in the real world.

The history of Web use consisted of content that the user had read on the Preview channel and texts that the user had written to other services in the My Life Assist Service. The history was expressed as a combination of literal texts and time stamps.

History from the real world consisted of a combination of locations and time stamps obtained by the mobile phone. A location was obtained by built-in GPS receiver or locator in the mobile phone that used radio waves from base stations.

2) Content for Recommendations: Content was recommended to each user in text form. It was desirable that each text, such as a description of a shop, was long enough to allow differentiation and to articulate groups of messages. It was also desirable that each text had corresponding locations in the real world so content could be selected based on location.

3) Converting Personal History to Text: The personal history on the Internet was acquired as text, while the real-world history was acquired as locations in the form of combinations of latitude and longitude. Most of the personal history consisted of location data, because texts were only obtained when the user used the service, while locations were obtained automatically.

Therefore, the system should discover the user's habits from a history of locations. It was presumed that the character of a location was described in terms of the content located in the area; the system converted a location to text from the associated content. In addition, the context-based recommender system did not aim to provide content around the user's location like existing location-based services. The system did not record locations, and did not recommend content based on the locations that the user often visited.

B. Articulation

As mentioned in Section III, the context-based recommender system generated graphs for both the personal history and the content by clustering based on similarities of expressions and linking based on context in the data. This graph generation was called articulation. In this paper, `articulation' means the decomposition of text

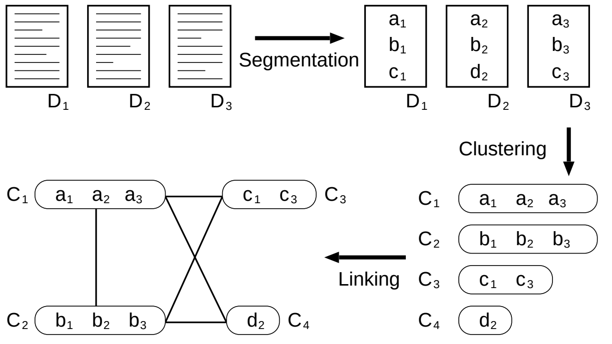


Figure 4. Example of articulation

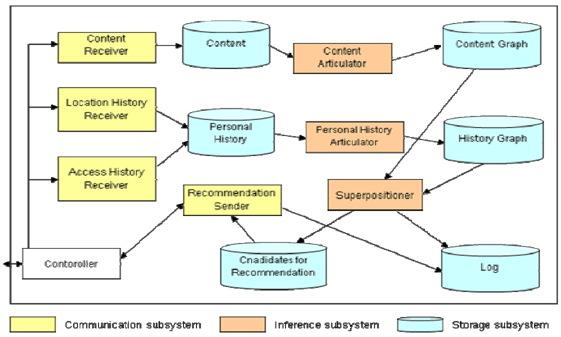


Figure 5. Components of the context-based recommender system

information into fragments by cutting and organizing the information into chunks connected with the writer's context.

In the articulation phase, the recommender system segmented each text of the personal history and the content. Next, the system grouped these text fragments into clusters. Then, it linked clusters that included text fragments from the same text.

In this way, graphs were generated from each personal history and the content. Each graph was organized from clusters that were found by a computer, based on similarities of text expressions and links based on the context of the expressions.

1) Example of Articulation: As shown in Figure 4, text fragments a_n, b_n, c_n, d_n were obtained from text D_1, D_2, D_3:

$D_1 = \{a_1, b_1, c_1\}$
$D_2 = \{a_2, b_2, d_2\}$
$D_3 = \{a_3, b_3, c_3\}$,

and they were clustered as follows:

$C_1 = \{a_1, a_2, a_3\}$
$C_2 = \{b_1, b_2, b_3\}$
$C_3 = \{c_1, c_3\}$
$C_4 = \{d_2\}$.

C_1 and C_2, C_1 and C_3, C_1 and C_4, C_2 and C_3, C_2 and C_4 included text fragments that were generated from the same text, so links were set between them (Figure 4). These links indicated contexts in original texts.

C. *Superposition*

The graphs for each user's personal history and the content were generated in the articulation phase. In the superposition phase, the recommender system looked for an overlap between the graph of a user's personal history and the graph of the content, extended the overlap together with links in the graph of the content, and selected items to recommend from the extended area.

Nodes of the graphs could be represented as word vectors because the graph was a group of text fragments. First, the system searched for similar nodes based on similarities between nodes represented as word vectors. The nodes found in this step in the content graph were named the primary nodes. Items in the primary nodes corresponded to the personal history. Next, the system followed links from the primary nodes, selected nodes that were connected to the primary nodes in the graph of the content, and called them secondary nodes. The secondary nodes were close to the personal history. Thus the system could extend overlaps of personal history and content. Then, the system selected items corresponding to the fragments as candidates for recommendation.

IV. Implementation

A. *Components of the Recommender System*

Figure 5 shows the components of the context-based recommender system and their relationships. The components of the system are the system controller, the communication subsystem (content receiver, personal location history receiver, personal access history receiver, recommendations sender), the inference subsystem (content articulator, personal history articulator, superpositioner), and the data storage subsystem (content storage, personal history storage, content graph storage, personal history graph storage, recommendation candidates storage, log storage). The communication subsystem and the inference subsystem both reference information stored in the data storage subsystem and work in parallel.

B. *Capturing and Storing Personal History*

Personal history was stored in the Service platform, which organized the histories into the desired format every three hours. The receiver subsystem in the recommender system received organized personal history from the platform at intervals of three hours or more.

Locations in the real world included in the personal history were converted into text that represented nearby content as mentioned in Section III-A.3. Items that had locations within a selected distance around a user's history of locations were considered content related to the user.

The history of Internet use, which consisted of items that the user had read on the Preview channel and text that the user wrote to other services in the My Life Assist Service, was considered to reflect a user's interests more directly than items related to the user. Therefore the receiver subsystem could give some weight to each history, and stored them in the personal history storage.

C. Articulation of Personal History and Content

The context-based recommender system aimed to extend the user's interests by providing enhanced context to the user's personal activities and augmented the content by discovering and extending the overlaps of personal activities and the content. First, the system articulated personal history and content for recommendations.

In the articulation phase, a graph structure of information was extracted, with these chunks as nodes and links as connections. An articulation process has three steps: segmentation that decomposes information into small fragments; clustering that groups fragments into clusters; and linking that connects clusters.

The system iterated articulation for 110 categories with 30,000 items selected from each category using the most recent first strategy.

1) Segmentation: In the segmentation phase, the recommender system segmented each text of the personal history and the content by windowing with a window size of L and the overlap $L/3$. Smaller values of L increased the number of branches in the graph; larger values reduced the number of branches [6].

There are text segmentation methods that analyze text expressions [7], [8]. However, the texts in the Preview channel were created without any prescribed rules by many people, and because the system applied clustering to text fragments in the next step, windowing was employed to split text.

2) Clustering: In this phase, the system converted fragments into vectors of words using a morphological analysis system. It used MeCab[3] as a morphological analyzer and collected nouns and unknown words. It generated vectors of words with the likelihood ratio[9], [10] between a fragment and each word in a fragment. Then it clustered them using hierarchical Bayesian clustering (HBC)[11].

Tfidf is generally used to generate vectors of words from documents, but the effect of word frequency is large and high-frequency words are not excluded [12]. Thus we introduced the likelihood ratio to generate word vectors. The common clustering methods for text data such as Ward's method tend to generate a few very large clusters that include half of all data, but HBC tends to generate clusters of similar size.

If a very large cluster is generated, almost all clusters have links to that cluster. It means that there is less variation of context in the graph structure of information [6]. Therefore we introduced HBC for the articulation.

3) Linking: Generated clusters are groups of text fragments, which are segmented from a text. Fragments from the same text can be distributed to several clusters. Thus, the system made links between clusters that included text fragments from the same text.

The strength of a link $l(i, j)$ from cluster C_i to cluster C_j was calculated using Jaccard's coefficient. That is, if $|C_k|$ is the number of original text fragments included in C_k:

$$l(i, j) = \frac{|C_i \cap C_j|}{|C_i \cup C_j|}.$$

In this way, graphs for both the personal history and the content were generated.

Clustering was performed based on similarity between text fragments, so clusters were groups with similar information generated by the system. On the other hand, linking was performed based on the number of fragments from the same text, so links indicated the context of the text creator's intention.

D. Superposition of Personal History and Content

The recommender system looked for an overlap between the graph of a user's personal history and the graph of the content, extended the overlap together with links in the graph of the content, and selected candidates for recommendation from the extended area.

1) Selecting Categories: The content was articulated for each of the 110 categories; the system then selected categories to recommend. It generated word vectors of all fragments of a user's personal history by the method mentioned in IV-C.2. It also generated the same kind of vector for each category of the content, and calculated the similarities the user's personal history and each category based on the similarities of these vectors.

The real-world part of the personal history was part of the content, converted from locations that related to items. Categories that consisted of many items with locations would appear frequently in personal histories. On the other hand, categories that had few locations would not appear often. Therefore, categories that included many locations would tend to be chosen as candidates for recommendation.

However, the Preview channel required the recommender system to recommend all categories. We therefore set weights for similarity between personal history and categories. The weight was N_A/N_P, where N_A was the number of items in a category and N_P was the number of items with locations in the category. The system selected categories as candidates based on weighted similarities.

2) Selecting Clusters and Content: The system generated word vectors from each cluster in the graph of each category that was selected as a candidate for recommendation. It also generated vectors from each cluster in the user's personal history.

It calculated similarities between each cluster in the graph of personal history and each cluster in the graph of selected categories of the content. It selected the combination of most similar clusters in the personal history and the content. The personal history cluster was named the base cluster, while the category cluster was named the main target cluster.

Next, the system selected three clusters that linked to a main target cluster based on the strength of the link $l(i, j)$ for a category. These clusters were named subtarget clusters, and formed the extended area of recommendation along with the context of the content.

[3] http://mecab.sourceforge.net/

The system generated word vectors of text fragments from the base cluster, the main target cluster, and the subtarget clusters. It calculated similarities between the base cluster and the fragments included in the main and subtarget clusters, to find K similar fragments. Then it stored items that had been segmented to these fragments in the recommendation candidate storage. It gave the recommendation candidate rank $RC_{c,k}$ to an item when it was included in the cth candidate category and had the kth largest similarity to the base cluster:

$$RC_{c,k} = K\,(c\text{-}1) + k.$$

E. *Choosing Recommendations*

Before sending recommendations to a user, the system selected them from the available candidates based on the location of the user. When a candidate had a location, the recommendation order $R_{c,k}$ was determined by (1); d was the distance in the real world between the user's location and the item and D_R was a constant that indicated how the distance in the real world affected the rank. On the other hand, if a candidate did not have a location, $R_{c,k}$ was determined by (2) with a constant value D_F :

$$R_{c,k} = RC_{c,k} + d\,D_R \quad (1)$$

$$R_{c,k} = RC_{c,k} + D_F \quad (2)$$

First, the system selected four categories to recommend with high recommendation ranks $R_{c,k}$. Next, it selected recommendations from each category based on $R_{c,k}$. In this way, four recommendations for each of the four categories were chosen and they were sent to the user's mobile phone via the My Life Assist Platform.

V. Experiments

A. *Outline of Service Operation*

The My Life Assist Service that included our context-based recommender system operated as a field experiment from December 2007 to February 2008. Figure 6 shows the weekly total of users and captured activities in the My Life Assist Service. In the last week of the experiment, the context-based recommender system obtained 196 users' personal histories. Activities obtained as personal histories over the same period totaled 601,773, including 360 accesses to the content via the Internet-access function of a mobile phone. Thus, 99.94% of activities were locations of users.

B. *Effect of Extending Context*

Table I shows the number of Web pages recommended and the number accessed that were selected from the primary nodes and the secondary nodes by 50 users in the last week of the field experiment. The access ratio for content selected from the secondary nodes was higher than that from the primary nodes by about 20%. Thus, we can say that the aim of the context-based recommender system, to recommend nonconverged contents and shift with the user's interest, was almost achieved.

VI. Summary

We have described a recommender system that recommended content based on personal histories. These

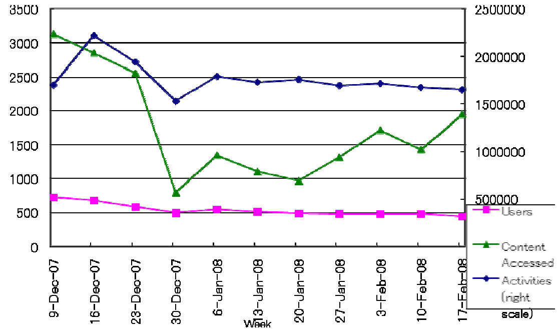


Figure 6. Weekly total of users and logs

TABLE I.
Number of Contents Recommended and Accessed (Feb.23-20, 50users)

Source	Recommended	Accessed	Ratio
Primary	9,071	72	0.79%
Secondary	5,898	58	0.98%
Total	14,969	130	0.87%

personal histories were collected by personal mobile phones, and consisted of a user's locations, a log of accesses to the Internet, and the user's contributions to the Internet. The system operated from December 2007 to February 2008 as a part of the *My Life Assist Service* by NTT DOCOMO, Inc. It recommended content from a user-generated Q&A Web site and a restaurant information Web site.

Because the service supported ordinary personal life via a mobile phone that was always with the user, the recommender system aimed to select content that was not converged, but varied.

We thought that the content expressed part of its creator's intention, and was located in a context. The recommender system found overlaps between the context of a Web page and the context of the user, extended the overlap along with the context of the content, and recommended Web pages from this extended area. In this way, the system recommended content from different aspects of the user's personal history.

In the experiment described in this paper, the recommender system extracted contexts of a Web page from the material recommended to users. However, it would be preferable for content to belong to various contexts. Clearly, content should consist not only of simple advertisements, but also background information that would interest a user. For example, if a Web page for a shop was selected using multiple contexts, not just suggestions from the shop but also information about the thoughts that the owner of the shop had had, the recommender system could make recommendations that may allow the user to enjoy visiting the shop more.

Content on the Internet is often created to attract public attention, sometimes to improve ranking in a search engine listing. However, to move beyond connecting a person to information by searching the Internet using key words, a system must create content in the context of the intentions of the creator that accords with the context of the user's personal history. A recommender system that finds the overlap between them, extends that overlap in the context of the content and presents the extended content, will enhance the user's experience. Then people will again be able to enjoy waves of information.

ACKNOWLEDGMENT

This research was a part of the *My Life Assist Service* by NTT DOCOMO, Inc. as a part of the *Information Grand Voyage Project* promoted by the Ministry of Economy, Trade and Industry in Japan. The authors would like to thank Dr. Koji Abe, AdIn Research, Inc., TAMON Co., Inc., NEC Corporation, Kanshin!, Inc., Kadokawa Cross Media Co., Ltd., and NTT Resonant Inc.

REFERENCES

[1] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[2] P. Resnick and H. R. Varian, "Recommender systems," *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.

[3] M. Hirose, "Lifelog and virtual timemachine," *Journal of the Japan Society of Mechanical Engineers*, vol. 109, no. 1056, pp. 892–895, 2006.

[4] S. M. McNee, J. Riedl, and J. A. Konstan, "Being accurate is not enough: how accuracy metrics have hurt recommender systems," in *Proc. SIGCHI '06 Conference on Human Factors in Computing Systems*, 2006, pp. 1097–1101.

[5] C. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proc. of the 14th International Conference on World Wide Web*, 2005, pp. 22-32.

[6] K. Tanaka and A. Takasu, "Topic change extraction from problem solving records," in *Proc. of the 8th World Multi-Conference on Systemics, Cybernetics and Informatics*, 2004.

[7] J. Allan, T*opic Detection and Tracking: Event-based Information Organization.* Kluwer Academic Publishers, 2002.

[8] M. Hearst, "Texttiling: Segmenting text into multi-paragraph subtopic passages," *Computational Linguistics*, vol. 23, pp. 33–64, 1997.

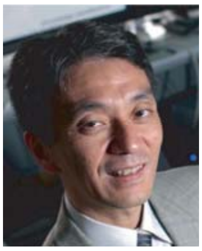
[9] T. Dunning, "Accurate methods for the statistics of surprise and coincidence," *Computational Linguistics*, vol. 19, pp. 61–74, 1993.

[10] M. Harada, S. Sato, and K. Kazama, "Finding key people and visualizing their relationships on the web," in *Information Processing Society of Japan SIG Notes, DBS-130/FI-71*, 2003, pp. 17–24.

[11] M. Iwayama and T. Tokunaga, "Hierarchical bayesian clustering for automatic text classification," in *Proc. 14th International Joint Conference on Artificial Intelligence*, 1995, pp. 1322–1327.

[12] A. Takano and S. Nishioka et al., "Development of the generic association engine for processing large corpora," in *19th IPA Technical Forum*, 2000.

Katsuaki Tanaka received his MS degree in informatics from Waseda University, Japan, in 2000. He is currently an assistant professor at the University of Tokyo. His research interests include natural language processing and knowledge management systems.
Mr. Tanaka is a member of JSAI and IPSJ.



Koichi Hori received B.Eng, M.Eng, and Dr.Eng. degrees in electronic engineering from the University of Tokyo, Japan, in 1979, 1981, and 1984, respectively.

In 1984, he joined National Institute of Japanese Literature, Japan, where he developed AI systems for literature studies. Since 1988, he has been with the University of Tokyo, Japan. He is currently a professor with School of Engineering, University of Tokyo. From September 1989 to January 1990, he also held a visiting position at University of Compiegne, France. His current research interest includes AI technology for supporting human creative activities, cognitive engineering, and Intelligent CAD systems.
Professor Hori is a member of IEEE, ACM, IEICE, IPSJ, JSAI, JSSST, and JCSS.

Masato Yamamoto received his MS degree in interfaculty information studies from the University of Tokyo, Japan, in 2004. He is currently a member of corporate sales division at NTT DOCOMO Inc. His research interests include natural language processing, knowledge management systems, and information retrieval.