Innovation Detection based on User-Interest Ontology of Blog Community

Makoto Nakatsuji, Yu Miyoshi, and Yoshihiro Otsuka

NTT Network Service Systems Laboratories, NTT Corporation,
9-11 Midori-Cho 3-Chome, Musashino-Shi, Tokyo 180-8585, Japan
{nakatsuji.makoto, miyoshi.yu, and otsuka.yoshihiro}@lab.ntt.co.jp

Abstract. Recently, the use of blogs has been a remarkable means to publish user interests. In order to find suitable information resources from a large amount of blog entries which are published every day, we need an information filtering technique to automatically transcribe user interests to a user profile in detail. In this paper, we first classify user blog entries into service domain ontologies and extract interest ontologies that express a user's interests semantically as a hierarchy of classes according to interest weight by a top-down approach. Next, with a bottom-up approach, users modify their interest ontologies to update their interests in more detail. Furthermore, we propose a similarity measurement between ontologies considering the interest weight assigned to each class and instance. Then, we detect innovative blog entries that include concepts that the user has not thought about in the past based on the analysis of approximated ontologies of a user's interests. We present experimental results that demonstrate the performance of our proposed methods using a large-scale blog entries and music domain ontologies.

1 Introduction

Blogs are becoming more popular for publishing and discussing interests among users who share interests between each other. In blog search, users can automatically pull blog entries from RDF Site Summary (RSS)[1] feed by entering keywords about their interests beforehand. Information-sharing systems of this type have the potential to enable users to expand their interests by browsing collected blog entries published by other users in blog communities.

However, information retrieval in current blog services relies only on keyword searches of blogs using Google or based on simple metadata such as that of an RSS. Moreover, there is no function to generate personalized searches easily, so users need to consider and enter search keywords that suit their own interests appropriately. Such a keyword search is time consuming and troublesome. Moreover, users cannot perform a keyword search if they do not understand what they want to search for to some degree beforehand. Thus, when keywords cannot be specified, information retrieval from blog entries often cannot be performed even if users might become interested in a topic.

[1] http://blogs.law.harvard.edu/tech/rss

To counteract the above problems, in the research on Adaptive Information Filtering (AIF) [2], the user profile is constructed cooperatively with a user, and recommendations based on the profile are offered. Making a user profile interactively beforehand is good for offering recommendations to users, as indicated by the high-accuracy performance of AIF. A common complaint about AIF is the user's task of making his/her own profiles, and a user often encounters known information many times because he/she cannot distinguish documents including new information in the recommendation results.

For filtering these redundant documents, researchers on novelty detection [7] define novelty as a document that includes new information that is relevant to a user profile. They extract relevant documents from a document stream. Then, they classify the documents as novel or not, and provide novelty documents to users. However, detecting novelty provides documents with information that includes concepts only in a user profile.

In this paper, we define *innovation* as new concepts which seem to be interesting to the user even though they are not included in a user profile. Then, we try to expand user interests significantly by recommending innovative information. Especially, we adopt innovation detection to blogs because they become a popular architecture of publishing and searching information that expands user interests.

For achieving above-mentioned purpose, we first construct user profile automatically as a user-interest ontology, which is a class hierarchy of user interests with interest weights. Then, we propose measuring the similarity of interest ontologies considering the degree of interest agreement to each class and instance. We apply our techniques to help users create a blog community by browsing innovative blog entries which include information unknown to users with a high probability of being interesting.

The specific contributions of this paper are the following.

- First, in order to analyze user interests in detail, we propose an automatic extraction of an interest ontology with an interest weight assigned to each class and instance. Bloggers are apt to describe their interests about topics in several service domains freely. Thus, we use blog entries for specifying user interests by introducing a template ontology, which is a domain ontology of each service. We classify user entries according to a template ontology, and remove classification mistakes by using class characteristics and continuity of descriptions about user interests. This mechanism of improving entry classification is one of the reasons for applying the ontology technique to our research.
- We propose measuring the similarity between interest ontologies that have interest weight. By introducing interest ontologies, we can help users create interest-based communities considering the width and depth of concepts of users' interests. Furthermore, we can calculate the similarity between ontologies more accurately than in previous ontology mapping techniques from the viewpoint of the agreement of the weights of user interests. Then, we can detect innovative blog entries for each user u by analyzing the classes

C of other users' ontologies that have a high similarity to the ontology of the user u though the interest ontology of user u lacks those classes C. This new approach of recommending innovative information is another reason for applying the ontology technique to our research.
– We describe a comprehensive set of experiments. Our experimental results are based on a large number of blog entries (1,600,000 entries of 55,000 users) and a music template ontology (114 classes and 4,300 instances). We confirm that our automatic ontology extraction and innovation detection have potential for creating a user-oriented blog community according to user interests. We also investigate the appropriate granularity of a community by analyzing the similarity of users' interests among the community extracted by our similarity measurements.

The paper is organized as follows. Section 2 introduces related works. Section 3 describes our automatic user-interest ontology-extraction, and Section 4 describes innovative blog-entry detection by our similarity measurement. Section 5 describes our experimental study, and Section 6 concludes this paper.

2 Related works

Many online content providers such as Amazon[2], offer recommendations based on collaborative filtering (CF) [5] which is a broad term for the process of recommending items to users based on the intuition that users within a particular group tend to behave similarly under similar circumstances. One advantage of previous CF techniques is that they can recommend relevant items that are different from those in a user's profile. However, they cannot detect innovative blog entries because only the similarity between user profiles based on instances such as selling items is measured. Therefore, CF often offers items that have the same concept to users. We want blog users to expand their interests by detecting innovative blog entries whose information is not included in the concepts (classes) of their interest ontology.

For applying a semantic approach to retrieving information from a blog, semblog [6] tries to construct a user profile using a personal ontology which is a manual construction of a users' classification of blog entries in a category directory of the ontology according to their interests. A category directory is built by users beforehand to construct an ontology-mapping-based search framework. However, manual ontology creation is a time-consuming and troublesome task for users, and applying a semantic ontology to a blog community is difficult. We automatically extract a user-interest ontology; thus, creating and updating ontologies is easy for users.

In researches of ontology mapping [1,3], similarity measurements considering approximation of classes and class topologies are proposed in [3]. In addition to class topology, we consider each user's weighted interest in each class and instance. Furthermore, in analyzing conjunctions in class topologies of ontologies

[2] http://www.amazon.com

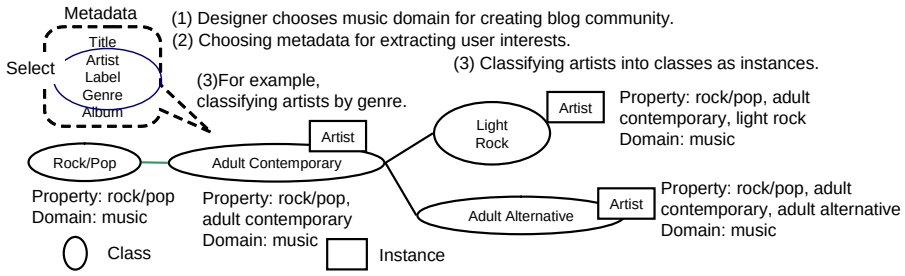


Fig. 1. Procedure for designing template ontology.

with high similarity scores, we detect innovative instances that a user does not have in his/her ontology, though other users have them with a high probability.

3 Interest ontology extraction

We first explain the template ontology design of each service domain such as those of content delivery services of music and movies and then describe an automatic method of extracting interest ontologies.

3.1 Design for template ontology

We use OWL (Web Ontology Language) [4] for describing a template ontology. We can express a domain ontology in detail using OWL. However, the generation and spread of a detailed ontology is obstructed because users have difficulty of designing it. Therefore, we design template ontologies as lightweight ontologies that only use a hierarchical relationship among the classes and a property description restricts the succession condition of a class hierarchy. Then, we automatically extract an interest ontology by classifying user blog entries into template ontologies without user intervention in Section 3.2.

As shown in Fig. 1, first, the ontology designer chooses a service domain for extracting user interests. Then, the designer chooses metadata that reflects user interests. In a music domain, the designer chooses metadata of genres or artists, considering the exsisting community is generated with such metadata. Finally, the designer chooses metadata as a restriction property of a class hierarchy and classifies other metadata as instances of classes. For example, the designer chooses genres as a property and classifies artists as instances of classes. In this way, we distinguish classes from instances and define the characteristics of classes based on the restriction properties of a class hierarchy and classified instances. We make use of these class characteristics to improve the accuracy of interest ontology generation in Section 3.2.

The service designers only has to construct a template ontology with the intended domains and gradually increase the number of ontologies along with

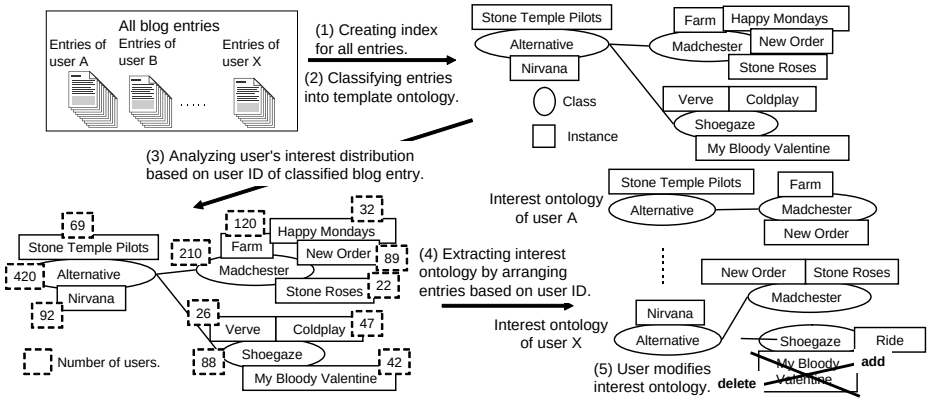


Fig. 2. Procedure for generating interest ontologies.

expanding the service. Designers also should adjust granularity of the end classes for reflecting user interests in detail. Fortunately, content directories such as goo music[3] set granularity in detail for users to browse contents according to their interests. Therefore, we first construct template ontologies according to these directories and evaluate the granularity through the analysis in Section 5.

3.2 Interest ontology generation algorithm

We explain our interest ontology generation algorithm by analyzing the interest distribution of users, as shown in Fig. 2.

Basic ontology generation algorithm First, we describe the basic ontology generation algorithm (BOGA) as follows.

(1) First, we make index files for all blog entries collected through the ping server. Here, we assume that collected blog entries have a unique user ID.

(2) Second, we classify all collected blog entries into a template ontology. We classify blog entry E_i into class C_i if there is a name attribute value of C_i in E_i. We also classify blog entry E_i into instance $I_i(\in C_i)$ if there is a name attribute value of I_i in E_i. We permit the blog entries to be classified into two or more classes. For example, consider the template ontology in Fig. 2. We classify the blog entry into instance "Happy Mondays" of class "Madchester" when there is a "Happy Mondays" character string in the description in the blog entry.

(3) Then, we measure the number of interested users in each instance of C_e, which is one of the end classes in the template ontology. On calculating the number of interested users, we count the number of users as one, even if the same user is describing the same instance or class in two or more blog entries.

[3] http://music.goo.ne.jp/

We calculate the number of interested users in class C_e by obtaining the number of interested users in all instances in C_e and in class C_e. Thus, the interested user distribution in the domain can be measured by recurrently counting the number of users from C_e to the root class C_r.

(4) Next, by extracting only the classification results about a user ID from all classification results, we can extract an interest ontology for this user ID. In Fig. 2, we can extract an interest ontology of user A when the blog entries of this user describe instances of "Stone Temple Pilots", "New Order", and "Farm".

(5) Finally, the user inspects and updates the interest ontology according to their interests. Furthermore, we can develop a template ontology that is more suitable by merging this modified information into a template ontology.

Ontology filtering algorithms For example, BOGA classifies blog entries that describe "Farm", which means an agricultural farm, into the instance "Farm" of class "Madchester". For filtering these mistakes caused by words with several meanings, we make use of the following characteristics such as class relationships in ontologies and durability of user interests in a blog.

- Instances that belong to the same class have the same characteristics.
- Adjacent classes have similar characteristics. Instances of those classes also have similar characteristics.
- User interests that continue for a certain period and describe an interest for two or more days.

We propose two filtering algorithms FA1 and FA2. First, we explain FA1.

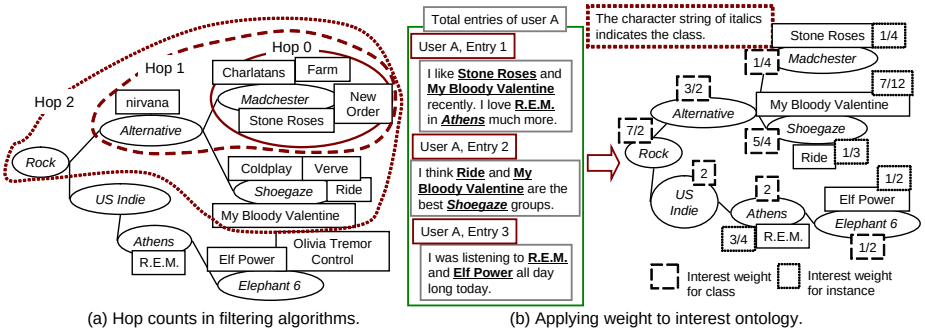
Filtering algorithm 1 We subdivide procedure (2) of BOGA for performing FA1.

(2-1) When the name attribute value $n(I_i)$ of instance $I_i(\in C_i)$ is described in blog entry E_i, FA1 checks whether a name attribute value of an instance of the same class (concept) $I_k\{(I_k \in C_i) \cap (I_k \neq I_i)\}$ or C_i is described in all blog entries that the user accumulates. We call instances I_k and C_i classification decision elements(CDEs).

(2-2) Entry E_i is classified as mentioning instance I_i when there is a description of CDEs, and not classified in I_i when there is no description. In Fig. 3, when the description of "Farm" exists in E_i, and "New Order" is described among all accumulation blog entries of a user, E_i is assumed to be a blog entry about instance "Farm" of "Madchester" and classified.

Filtering algorithm 2 We propose filtering algorithm 2 (FA2) whose classification is stronger than FA1. In procedure (2-1) of FA1, FA2 checks whether CDEs are described in blog entry E_i. Then, blog entry E_i is classified in I_i when there is a description of CDEs, and not classified in I_i when there is no description.

Adjusting the range of CDEs We give a mechanism that adjusts the range of CDEs by using the class hierarchy. We consider that descriptions of classes



(a) Hop counts in filtering algorithms. (b) Applying weight to interest ontology.

Fig. 3. (a) hops in filtering algorithms, and (b) applying interest weight to ontology.

and instances of interest often appear with instances of the neighboring classes. We add a new adjustment parameter, hop, which defines the range of CDEs. In Fig. 3-(a), we assume brother classes, the grandfather class, and instances that belong to each of CDEs when there are two hops from end classes.

3.3 Introducing interest weight to ontology

In addition, we introduce the interest weight as a parameter that shows the degree of a user's interest in each class and instance of an interest ontology. By using this parameter, we can create a community among users who have almost the same degree of interest in the same classes or instances.

Here, we define interest weight, as shown in Fig. 3-(b). First, the interest weight of every blog entry is one. Second, if there are $N(E_i)$ kinds of name attribute values of interest classes and instances that appear in blog entry E_i, the interest weight of each class and instance in E_i becomes $1/N(E_i)$. Third, when we define the set of all accumulation blog entries of a user as E, the interest weight $S(I_i)$ of each instance I_i is $S(I_i) = \sum_{(I_i \in E_i)}^{|E|}(1/N(E_i))$, and the interest weight $S(C_i)$ of each class C_i is $S(C_i) = \sum_{(C_i \in E_i)}^{|E|}(1/N(E_i)) + \sum_{I_i \in C_i} S(I_i)$. Fourth, the interest weight of the instances is reflected in that of the class that includes the instance. The interest weight of the classes is reflected in that of the super class. For example, in Fig. 3-(b), we give the interest weight of instance "Elf Power" as $1/2$, instance "R.E.M." as $1/4 + 1/2 = 3/4$, class "Elephant 6" as $1/2$, and class "Athens" as $1/2 + 3/4 + 1/2 + 1/4 = 2$.

4 Detecting innovative blog entries using similarity measurements

We propose measuring the similarity between ontologies considering interest weight. Then, we describe innovative blog-entry detection and community creation support based on the analysis of interest ontologies with high similarity.

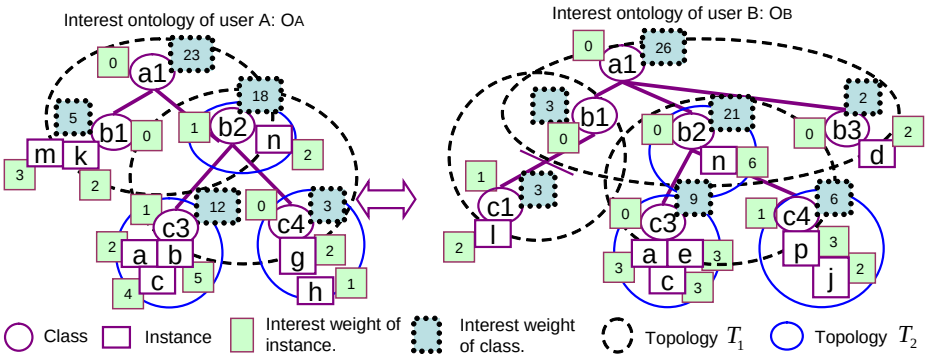


Fig. 4. Measuring similarity based on the degree of interest agreement.

4.1 Interest-weight-based similarity measurement

We now explain our similarity measurement in detail by using Fig. 4.

We first define terminologies. We give interest ontology O_A of user A and O_B of user B, topology T_1, which is defined as the relation between a class and subclasses, and topology T_2, which is defined as the relation between a class and instances. Furthermore, we define common classes of both ontologies as C_i, and common instances as I_i. In particular, we define common class set, $C(T_1)$, as that which characterizes topology T_1, and common class set, $C(T_2)$, as that which characterizes topology T_2. For example, in Fig. 4, $C(T_1)$ has common classes $a1$ and $b2$, and $C(T_2)$ has common classes $b2$, $b3$, and $c4$. We also give the degree of interest agreement of common instance I_i as $I(I_i)$, that of common class C_i as $I(C_i)$, and that of common topology created by common class C_i as $I_t(C_i)$.

In [3], the authors calculate the similarity between ontologies considering the degree of similarity between class topologies T_1. In addition, we take the following ideas from the view point of creating a user-interest-based community.

- Evaluating the degree of interest agreement between C_is and I_is as a smaller value of interest weight. This idea is for filtering users who only enumerate a lot of instances in an entry, and creating a community among users who have similar or larger interest weight values from the viewpoint of each user.
- Separately treating topologies T_1 and T_2 because we consider that T_1 reflects the width and depth of a user's interests and T_2 reflects the objects in which users are interested.
- Achieving a low computational complexity by generating the class schema of user-interest ontologies accroding to that of template ontologies. This is important for ontology mapping to adopt large-scale dataset of blog community such as that of our experiments in Section 5.

(1) We analyze classes common to O_A and O_B and extract common classes which belong to $C(T_1)$ and $C(T_2)$.

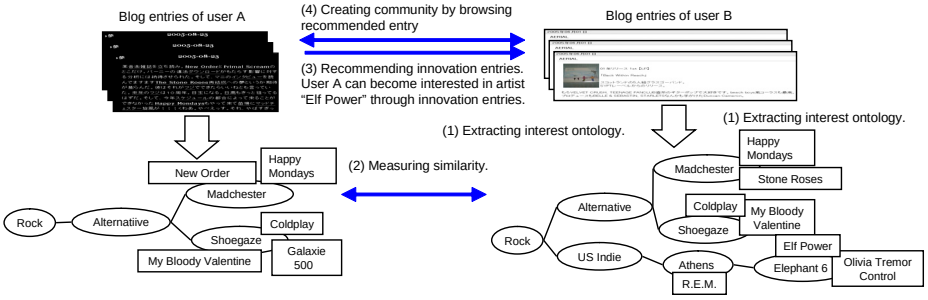


Fig. 5. Community creation service of recommending innovative blog entries.

(2) When common class C_i has common instance I_i between ontologies, we assign the smaller value of the interest weight of common instances I_i to $I(I_i)$. For example, $I(a)$ is 2.

(3) Similarly, we assign the smaller value of the interest weight of common class C_i to $I(C_i)$. For example, $I(b1)$ is 3.

(4) We define product sets of subclasses of C_i, which are common to a class set, as $N(C_i)$, and the set union of subclasses of C_i among $C_i \in C(T_1)$ as $U(C_i)$. For example, $N(a1) = \{b1, b2\}$ and $U(a1) = \{b1, b2, b3\}$. Then, we give $I_t(C_i)$ as $\frac{\sum_{C_j \in N(C_i)} I(C_j)}{|U(C_i)|}$. For example, $I_t(a1)$ is given by $(3+18+0)/3 = 7$. Thus, we obtain degree of interest agreement $S(T_1)$ of $C(T_1)$ as $\sum_{C_i \in C(T_1)} I_t(C_i)$. In Fig. 4, $S(T_1) = (3+18+0)/3 + (9+3)/2$.

(5) We also define an instance set of C_i in ontology O_A as $I_A(C_i)$, and an instance set of C_i in ontology O_B as $I_B(C_i)$ among $C_i \in C(T_2)$. Then, we give $I_t(C_i)$ as $\frac{\sum_{I_i \in C_i} I(I_i)}{|I_A(C_i) \cup I_B(C_i)|}$. For example, $I_t(c3)$ is given by $((2+0+3+0)/4) = 5/4$. Thus, we assign the degree of interest agreement $S(T_2)$ of $C(T_2)$ as $\sum_{C_i \in C(T_2)} I_t(C_i)$. In Fig. 4, $S(T_2) = 2/1 + 5/4 + 0$.

(6) By using evaluation function $f(X)$ corresponding to the relative degree of importance of a topology, we finally assign the similarity score between ontologies $S_O(AB)$ as $S(T_1) + f(S(T_2))$.

4.2 Innovative blog-entry detection

We adopt our similarity measurement to innovative blog-entry detection.

(1) We calculate the similarity between the ontology of user A and ontologies of other users in set U. By using the heuristic threshold X, we derive X users who have a high similarity to user A as an interest-sharing community G_U.

(2) Then, we analyze difference instances between the ontology of user A and ontologies of G_U. We also define a parameter, degree of innovation, which indicates how many hops we need to get from difference instances of an ontology of G_U to the class of the ontology of user A. In Fig. 5, we need 3 hops to go from

difference instance "Elf Power" of ontology of user B to class "Rock" of ontology of user A. By recommending blog entries with a high degree of innovation, users may significantly expand their interests. Otherwise, users may receive new concept with a low degree of innovation comparatively more acceptable.

(3) Finally, we extract innovative instances G_I, which user A does not have, even though users in G_U have with a high possibility, and recommend innovative blog entries about G_I for user A with innovation degree.

Fig. 5 depicts an example of our community creation. We can analyze whether a user who is interested in instance "Happy Mondays" of class "Madchester" and so on has a possibility to become interested in instance "Elf Power" of class "Elephant 6". By browsing blog entries concerning these innovative instances, users expand their interests and share interests with each other.

5 Experimental results

We now present experimental results that show the performance of interest ontology extraction and innovative blog-entry detection.

5.1 Datasets and methodology

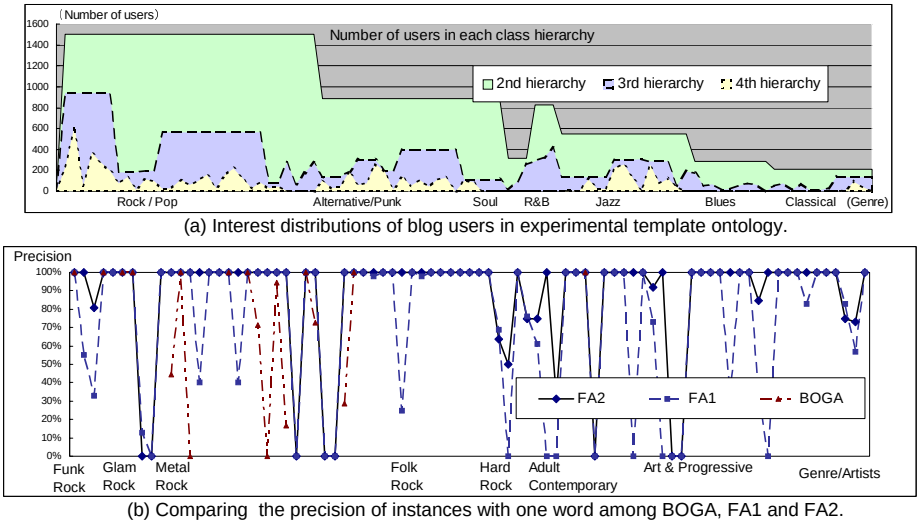
We evaluated the performance of our proposed methods based on the large-scale blog portal Doblog[4], which has 1,600,000 blog entries of 55,000 users. We also used the template ontology of the music domain, as shown in Fig. 2, which was created referring to public information about web portals such as goo music. Our experimental template ontology contains 114 classes as genres and 4,300 artists as instances, and each class and instance have two or more name attribute values. For example, the instance "R.E.M." has the name attribute values of "R.E.M." and "REM". Thus, we gave 7,600 name attribute values to 4,300 instances.

For evaluating accuracy, we defined correct answers as blog entries that have descriptions of classified classes or instances and evaluated the generated interest ontology by using precision and recall in classified results. In this paper, precision means the proportion of correct answers in classified results and recall means that of correct answers in all blog entries. When the recall is high, extracted interest ontologies cover user interests better. However, when the precision is lower, created interest ontologies include classified mistakes, and innovation detection for the user becomes unreliable. Thus, achieving high precision is indispensable. In evaluation, we adopted filtering algorithms to instances with one word such as "police", because we considered one word has a high possibility of having several meanings. For generating index files of blog entries, we used Namazu[5].

5.2 Measuring interest distributions of blog users

Graphs of user distributions in the music domain of our experiment are depicted in Fig. 6-(a). There are about 200 users, even in end classes. By checking the blog

[4] http://www.doblog.com
[5] http://www.namazu.org/



(a) Interest distributions of blog users in experimental template ontology.

(b) Comparing the precision of instances with one word among BOGA, FA1 and FA2.

Fig. 6. Experimental results of user distributions and ontology extraction.

entries classified in end classes, we confirmed that these blog entries frequently have unique words, which describe the features of these classes. For example, blog entries classified into the end class "Death Metal" have the phrase "death voice" with a high probability. This is because the end classes in our template ontology have an appropriate granularity to extract the feature of the blog entries classified into these classes. The granularity of end classes is important because it affects whether we can determine if a user is interested in the community.

5.3 Measuring performance of extracted interest ontology

We evaluated the accuracy of FA2 by checking 1/4 of classified blog entries, which were randomly selected. As shown in Table. 1-(a), the achieved precision is higher than 90% with a high recall of 80%. Thus, our filtering algorithm is effective for generating suitable user-interest ontologies.

5.4 Comparing filtering algorithms

Then, we compared BOGA and filtering algorithms by randomly checking 1/4 of the blog entries, which were classified into instances with one word.

Graphs of the precision of BOGA, FA1, and FA2 over 83 instances, which were randomly selected among 827 instances with one word, are shown in Fig. 6-(b). The accuracy between BOGA and filtering algorithms is compared in Table. 1-(b). These results indicate that precision improves in the order of BOGA, FA1, and FA2, and recall decreases significantly in FA2, even though FA1 drops only

Table 1. Experimental results of our ontology extraction and innovation detection.

(a) Accuracy of extracted interest ontology (FA2, hop 2).

Precision	Recall
94.9%	80.3%

(b) Comparing accuracy of instances with one word.

	FA2	FA1	BOGA
Precision	70.0%	57.9%	18.9%
Recall	32.6%	93.0%	100.0%

(c) Comparing accuracy by changing hop counts.

	Hop 0	Hop 2	Hop 4
Precision	89.1%	91.0%	85.6%

(d) Recall of innovation detection.

	X=30	X=60	X=90
Recall	64.8%	76.7%	80.1%

(e) Comparing degree of innovation in recommendation lists.

Degree of innovation	0	1	2	3
Proportion	57.6%	15.2%	23.2%	4.0%

(f) Comparing degree of innovation in our detections.

Degree of innovation	0	1	2	3
Proportion	23.4%	23.1%	44.3%	9.2%

slightly from BA. For improving recall with high precision in FA2, we will add a method that checks for CDEs in the blog entries with a high probability of appearing these elements such as entries near each other in a time series.

Analyzing Fig. 6-(b) in more detail, there are eight instances in which the precision cannot be improved even with FA2, and they lower the overall precision. Then, we extracted instances in which the number of classifications increases by ten times or more when changing from FA2 to FA1. As a result, we extracted 28 instances and the precisions of 5 of those instances were 0. The reason is that they do not co-occur in the same blog entry with CDEs, even though the user was interested in them and described the name attribute value of these instances often. Thus, to improve the precision, deleting these instances from template ontology is effective.

We also evaluated the accuracy of FA2 based on the change in the hop number. Hop 2 is better than hop 0 with respect to the number of correct answers and precision, as shown in Table. 1-(c). However, hop 4 is lower than hop 2 in precision, although the number of correct answers is slightly better. That is because our template ontology has a large number of instances in end classes, and the relationship between end classes and super classes is closer than the relationship between super classes and grandfather classes. For example, end class "Acid Metal" has the super class "Metal" and grandfather class "Rock". In this case, the relationship between "Acid Metal" and "Metal" is closer than the relationship between "Metal" and "Rock". Thus, hop 2 has a better precision than hop 0 because hop 2 has many CDEs, and hop 4 has a lower precision than hop 2 because we consider CDEs in hop 4 as instances that are far from end classes.

5.5 Measuring performance of innovation detection

We evaluated innovative blog-entry detection. In the evaluation, we defined correct answers for each instance by referring to recommendation lists such as "you might like these artists" in a music portal like goo music. Designers of music portals in this evaluation manually defined artists (A_n) that are relevant to another

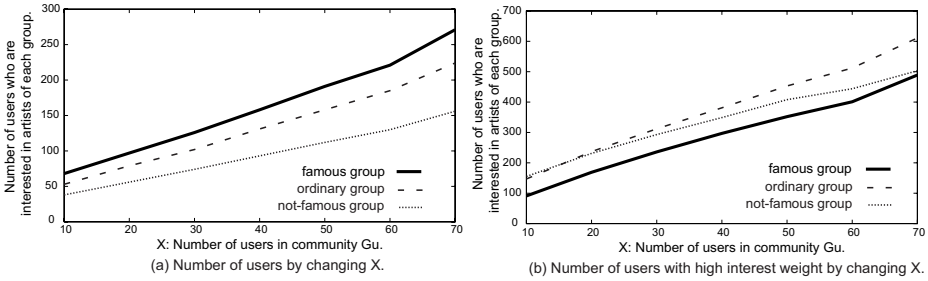


Fig 7. (a) number of users obtained by changing X. (b) number of users obtained that have high interest weight by changing X.

artist (A_i) for recommending relevant artists (A_n) to users who are interested in artist (A_i). Then, we evaluated our technique by checking the recall of 1/20 of 1503 users who were judged to be interested in the music domain of our template ontology. In this evaluation, recall means the proportion of correct answers in our recommended instances.

We evaluated recall in the change of X described in Section 4.2. Table. 1-(d) indicates that recall of our recommendation was about 80%. In particular, recall improves significantly when $X = 30 - 60$, even though $X = 90$ improves slightly from $X = 60$. This result indicates that we can extract innovative instances by only checking 60 high-rank interest ontologies among interest ontologies of 1503 users from the viewpoint of the user who receives the recommendation. Table. 1-(e) and (f) compare the proportion of degree of innovation in extracted instances between recommendation lists in a music portal and our detected instances. These results indicate that our technique detects instances with high degree of innovation more in number than recommendation lists.

5.6 Analyzing the suitable granularity of user-oriented community

We also investigated suitable number of users for creating a community. First, we selected a user among all users extracted by our template ontology and analyzed suitable granularities of G_U by changing parameter X described in Section 4.2. In this evaluation, we divided innovative instances G_I into 3 instance groups in order of the appearance rate of instances when we set X to 70: a famous group, an ordinary group, and a not-famous group. We calculated the number of users who are interested in the artists of each group by changing X from 10 to 70.

Graphs of the number of users who are interested in each group obtained by changing X are shown in Fig. 7-(a). Next, we focused on users who have a high interest weight in their interest ontologies. Graphs of the number of such users obtained by changing X are shown in Fig. 7-(b). A famous group is recommended to users in spite of changes in X in Fig. 7-(a). On the other hand, in Fig. 7-(b), a not-famous group is recommended most when X is 10, and a normal group

comes to be recommended gradually as X grows. This is because users with a high interest weight have a tendency of discussing not-famous instances, in spite of discussing famous instances. Furthermore, the number of users of each group increases suddenly when X is greater than 60. This is because the gap between a user's ontology and ontologies of G_u is larger when X is greater than 60, and instances with a low possibility of being interesting come to be recommended more often. From the results of Section 5.5 and 5.6, our innovation detection is effective according to detailed user interests when X is smaller than 60.

6 Conclusion

We proposed an interest ontology generation method and similarity measurement considering interest weight. Then, we adapted our technique to detect innovative blog entries in a blog community. We also performed large-scale experiments and confirmed that our techniques achieved automatic ontology extraction and detection of innovative blog entries with high accuracy.

We offer an experimental service DoblogMusic[6] for Doblog users and confirm the effectiveness of our innovative blog-entry recommendation method for creating a blog community by analyzing user access during a period of time.

Acknowledgments

In the verification of this research we used data from blog portal Doblog of NTT DATA Corporation. I wish to express my gratitude to the Doblog team and Hottolink Corporation, which pleasantly cooperated in offering the data and discussing the blog community creation service.

References

1. Doan, A., Madhavan, J., Domingos, P., and Halevy, A.: Learning to map between ontologies on the semantic web, in *The 11th International WWW Conference* (2002).
2. Godoy, D. and Amandi, A.: User Profiling in Personal Information Agents: A Survey, *Knowledge Engineering Review, Cambridge University Press* (2005).
3. Maedche, A. and Staab, S.: Measuring Similarity between Ontologies, *Proc. Of the European Conference on Knowledge Acquisition and Management - EKAW-2002. Madrid, Spain, LNCS/LNAI 2473, Springer*, pp. 251–263 (2002).
4. McGuinness, D. L. and v. Harmelen, F.: Web Ontology Language (OWL): Overview, W3C Recommendation, http://www.w3.org/TR/owlfeatures/ (2004).
5. O'Donovan, J. and Dunnion, J.: Evaluating Information Filtering Techniques in an Adaptive Recommender System, *Proceedings of Adaptive Hypermedia and Adaptive Web-Based Systems*, pp. 312–315 (2004).
6. Ohmukai, I. and Takeda, H.: Metadata-Driven Personal Knowledge Publishing., *International Semantic Web Conference*, pp. 591–604 (2004).
7. Zhang, Y., Callan, J. and, Minka, T.: Novelty and redundancy detection in adaptive filtering, *Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval*, pp. 81–88 (2002).

[6] http://music.doblog.com/exp/index