3B4-3

# PMI-IRの拡張によるユーザ行動モデルの階層関係抽出

## Extraction of Hierarchical Relation between User's Activity based on Enhanced PMI-IR

深澤 佑介[*1*2]　太田 順[*2]
Yusuke Fukazawa　Jun Ota

[*1]株式会社 NTT ドコモ　[*2]東京大学
NTT DOCOMO, Inc.　The University of Tokyo

We have been developing a task-based service navigation system that offers to the user services relevant to the task the user wants to perform. To realize automatic modeling of user's task-model from the web, in this paper, we describe the method estimate the hierarchical relationships present in the activity model with the lowest possible error rate, which is one of the research questions to the above goal. Concretely, we propose a method that divides the representation of activities into a noun part and a verb part, and calculates the mutual information between them by enhancing the idea of PMI (Pointwise mutual information). The result shows almost 80% of the hierarchical relationships can be captured by the proposed method.

## 1. Introduction

The mobile Internet is expanding dramatically, such as the number of subscribers and the volume of mobile contents. As the mobile Internet gains in popularity, information retrieval must be made easier and more efficient. Towards this goal, we proposed a task-based service navigation system[3][5] that supports the user in finding appropriate services. Naganuma et al. proposed a method for constructing a rich task-model that represents a wide variety of user activities in the real world. To use the task-model for service navigation, the user enters a task-oriented query such as "Go to theme park" and a list of tasks that match the query is sent to the mobile device. The user selects the most appropriate task and, in turn, the corresponding detailed sub-tasks are shown to the user. By repeatedly selecting a task and its sub-tasks, the user can clarify the demand or problem, and when the user reaches an end-node task, appropriate services associated with the selected task in the service DB are shown; a service is invoked by clicking its URI link.

The above task-model aims at modeling general real world activities that could be performed by the average mobile user. Existing task-models are mainly constructed by domain-experts, however, this approach suffers from narrow coverage and hinders the updating of the task-model.

Therefore, in this paper, we investigate the automatic modeling of users' real world activities from the web. We use the web as the resource because the web, especially user-generated contents such as blogs, include enormous volumes of recently updated information on users' daily activities in the real world. This allows mobile users to exploit the expressiveness of the semantic data from the Web for semantic IR.

### 1.1 Related works

Related works on learning task-models can be categorized into two types depending on what kinds of resources are used for learning, i.e. structured or unstructured data. Here, we describe the method that use unstructured data. Sabou et al. [6] proposed a method to extract the functionality of web services from the web. They use lexico-syntactic patterns to extract verbs and their objects as descriptions of functionality. For example, <find> <antigenic site> is identified as a lexical construct denoting a possible functionality in the bioinformatics domain. David [7] proposed to learn non-taxonomic relations between two concepts in a medical ontology (e.g. "high sodium diet" "is associated with" "hypertension"). So as to extract non-taxonomic relations, they focused on the extraction of domain and domain related verbs (e.g. breast cancer is caused by) by using lexico-syntactic patterns. He also propose PMI (Pointwise mutual information)[2][1] based co-occurrence analysis to filter noisy combinations of domain and domain related verbs. The proposed method showed high precision and recall in an evaluation test.

Our goal for the task-model is to support the user in navigating to contents that will satisfy the user's task by concretizing the user's request. For this goal, we have the following research questions: How to accurately acquire (i.e. low error rates) the hierarchical relationships between activities? Against this research question, the problem can be considered as the calculation of semantic distance between two concepts. In this paper, we enhance the idea of PMI (Pointwise mutual information)[2][1] based co-occurrence analysis, which is considered to be the best method for calculating the distance between two noun concepts, to calculate the distance between two tasks.

The rest of paper is organized as follows. Section 2 presents our proposed method for learning/extracting a comprehensive task-model from the web. Section 3 evaluates the effectiveness of the proposed method. Finally, our conclusions are presented in Section 4.

連絡先: Contact: fukazawayuu@nttdocomo.co.jp[*1], ota@race.u-tokyo.ac.jp[*2]

# 2. Extracting a task-model from the web

## 2.1 Light-weight task-model

We describe the task-model targeted in this paper (the light-weight task-model). The structure of the light-weight task-model is shown in Fig.1. This model consists of domains and activities. The top node of the model represents a domain; second level nodes indicate the activities identified by the processes associated with the domain. The third level represents the concrete activities for each activity in the 1st level. Note that the model sets only hierarchical relationships between activities. To automatically construct this task-model, we need three processes shown in the figure. Both process 1 and 2 have been described in [4], and we detail process 3 in this paper.

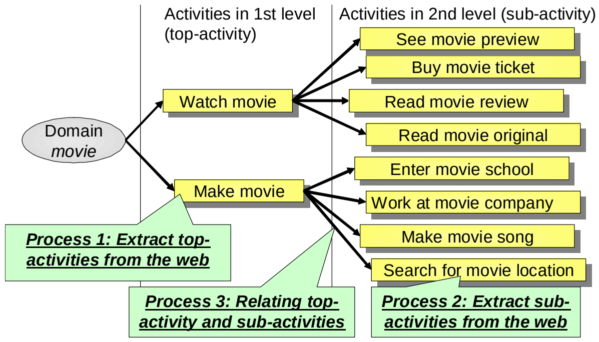
Activities in 1st level (top-activity)
Activities in 2nd level (sub-activity)
Domain movie
Watch movie
Make movie
See movie preview
Buy movie ticket
Read movie review
Read movie original
Enter movie school
Work at movie company
Make movie song
Search for movie location
Process 1: Extract top-activities from the web
Process 3: Relating top-activity and sub-activities
Process 2: Extract sub-activities from the web

図 1: Structure of light-weight task-model

## 2.2 Relating sub-activity to top-activity

This step figures out, from among the sub-activity candidates listed in process 1, which are correct and most strongly related to the top-activity. This step can be considered as the calculation of semantic distance between two concepts. Therefore, we extend the idea of PMI (Point-wise mutual information)[2][1] based co-occurrence analysis, which is considered to be the best method for calculating the distance between two noun concepts, to calculate the distance between two tasks.

Simply applying the idea of PMI to calculating the semantic distance between tasks yields mutual information $I(p;c)$ where $p$ and $c$ are the top-activity and sub-activity, respectively. The-activity that has the highest $I(p;c)$ is selected as the top-activity. $I(p;c)$ is calculated as follows.

$$I(p;c) = \{H(p) + H(c) - H(p,c)\} = pmi(p,c)$$

where $H(p)$ and $H(c)$ are the marginal entropies and $H(p,c)$ is the joint entropy of $p$ and $c$. $pmi(x,y) = \log\left(\frac{hits(x\mathrm{AND}y)}{hits(x)hits(y)}\right)$. We call this approach Method1.

There is, however, the following problem with Method1. As most $hits(c)$ and $hits(p)$ are expected to be small because task does not appear frequently in the resource on the web, $hits(p,c)$ falls to around 0, and the top-activity cannot be determined. That is, the relationship between activities acquired in Method1 is thought to exhibit high precision but very low recall.

In order to improve recall, we divide the representation of activity (top-activity or/and sub-activity) into a noun part and a verb part, and calculate the mutual information between them as separate entities. As the number of search results of noun/verb parts of activities are generally larger than that of the activity, the number of relationships to be calculated is expected to be increased. To investigate the effectiveness of task-division depending on which activity (top-activity or/and sub-activity) is/are divided, we develop following three patterns. In Method2, we divide only the sub-activity into noun part $c_n$ and verb part $c_v$, and calculate $I(c_n;c_v;p)$. In Method3, we divide only the top-activity into noun part $p_n$ and verb part $p_v$, and calculate $I(p_n;p_v;c)$. In Method4, we divide both top-activity and sub-activity into noun part and verb part, and calculate $I(p_n;p_v;c_n;c_v)$.

We calculate $I(c_n;c_v;p)$ of Method2 as follows:

$$\begin{aligned}
&I(c_n;c_v;p) = I(c_n;c_v) - I(c_n;c_v|p)\\
&= \{H(c_n) + H(c_v) - H(c_n,c_v)\}\\
&\quad -\{H(c_n|p) + H(c_v|p) - H(c_n,c_v|p)\}\\
&= \{H(c_n) + H(c_v) - H(c_n,c_v)\}\\
&\quad -\{H(c_n,p) - H(p) + H(c_v,p) - H(p)\\
&\quad -H(c_n,c_v,p) + H(p)\}\\
&= H(c_n) + H(c_v) + H(p)\\
&\quad -H(c_n,c_v) - H(c_n,p) - H(c_v,p) + H(c_n,c_v,p)\\
&= \{H(c_n) + H(p) - H(c_n,p)\}\\
&\quad +\{H(c_v) + H(p) - H(c_v,p)\}\\
&\quad -\{H(c_n,c_v) + H(p) - H(c_n,c_v,p)\}\\
&= pmi(c_n,p) + pmi(c_v,p) - pmi(c_n\mathrm{AND}c_v,p)
\end{aligned}$$

where $H(X|Y)$ and $H(Y|X)$ are the conditional entropies.

We calculate $I(p_n;p_v;c)$ of Method3 as follows:

$$\begin{aligned}
&I(p_n;p_v;c) = I(p_n;p_v) - I(p_n;p_v|c)\\
&= \{H(p_n) + H(p_v) - H(p_n,p_v)\}\\
&\quad -\{H(p_n|c) + H(p_v|c) - H(p_n,p_v|c)\}\\
&= \{H(p_n) + H(p_v) - H(p_n,p_v)\}\\
&\quad -\{H(p_n,c) - H(c) + H(p_v,c) - H(c)\\
&\quad -H(p_n,p_v,c) + H(c)\}\\
&= H(p_n) + H(p_v) + H(c)\\
&\quad -H(p_n,p_v) - H(p_n,c) - H(p_v,c) + H(p_n,p_v,c)\\
&= \{H(p_n) + H(c) - H(p_n,c)\}\\
&\quad +\{H(p_v) + H(c) - H(p_v,c)\}\\
&\quad -\{H(p_n,p_v) + H(c) - H(p_n,p_v,c)\}\\
&= pmi(p_n,c) + pmi(p_v,c) - pmi(p_n\mathrm{AND}p_v,c)\\
&= cons. + pmi(p_v,c) - pmi(p_n\mathrm{AND}p_v,c)
\end{aligned}$$

Here, we set $pmi(p_n,c)$ as a constant value, which does not have to be calculated, because $p_n$ represents the input domain and is common in all candidate pairs $p$ and $c$.

We calculate $I(p_n;p_v;c_n;c_v)$ of Method4 as follows:

$$
\begin{aligned}
& I(p_n;p_v;c_n;c_v) = I(p_n;p_v;c_n) - I(p_n;p_v;c_n|c_v) \\
& = H(p_n) + H(p_v) + H(c_n) + H(c_v) \\
& \quad -H(p_n,p_v) - H(p_n,c_n) - H(p_n,c_v) \\
& \quad -H(p_v,c_n) - H(p_v,c_v) - H(c_n,c_v) \\
& \quad +H(p_n,p_v,c_n) + H(p_n,p_v,c_v) \\
& \quad +H(p_n,c_n,c_v) + H(p_v,c_n,c_v) \\
& \quad -H(p_n,p_v,c_n,c_v) \\
& = \{H\,(p_n) + H(c_n) - H(p_n,c_n)\} \\
& \quad +\{H(p_n) + H(c_v) - H(p_n,c_v)\} \\
& \quad +\{H(p_v) + H(c_n) - H(p_v,c_n)\} \\
& \quad +\{H(p_v) + H(c_v) - H(p_v,c_v)\} \\
& \quad -\{H(p_n,p_v) + H(c_n) - H(p_n,p_v,c_n)\} \\
& \quad -\{H(p_n,p_v) + H(c_v) - H(p_n,p_v,c_v)\} \\
& \quad -\{H(p_n) + H(c_n,c_v) - H(p_n,c_n,c_v)\} \\
& \quad -\{H(p_v) + H(c_n,c_v) - H(p_v,c_n,c_v)\} \\
& \quad +\{H(p_n,p_v) + H(c_n,c_v) - H(p_n,p_v,c_n,c_v)\} \\
& = pmi(p_n,c_n) + pmi(p_n,c_v) + pmi(p_v,c_n) + pmi(p_v,c_v) \\
& \quad -pmi(p_n \mathrm{AND} p_v,c_n) - pmi(p_n \mathrm{AND} p_v,c_v) \\
& \quad -pmi(p_n,c_n \mathrm{AND} c_v) - pmi(p_v,c_n \mathrm{AND} c_v) \\
& \quad +pmi(p_n \mathrm{AND} p_v,c_n \mathrm{AND} c_p) \\
& = cons. + pmi(p_v,c_n) + pmi(p_v,c_v) \\
& \quad -pmi(p_n \mathrm{AND} p_v,c_n) - pmi(p_n \mathrm{AND} p_v,c_v) \\
& \quad -pmi(p_v,c_n \mathrm{AND} c_v) \\
& \quad +pmi(p_n \mathrm{AND} p_v,c_n \mathrm{AND} c_v)
\end{aligned}
$$

Here, we set $pmi(p_n,c_n)$, $pmi(p_n,c_v)$ and $pmi(p_n,c_n \mathrm{AND} c_v)$ as constant values, which do not have to be calculated, as $p_n$ represents the input domain and is common in all candidate pairs $p$ and $c$.

## 3. Evaluation

In this section, we evaluate the rate at which sub-activities are erroneously associated with top-activities and compare the error rates yielded by methods 1-4 in Section 2.2. The search engine Bing allows us to search contents for a specific domain by adding the site domain to the query. For instance, in order to search from blogs, we add "site:http://blogs.yahoo.co.jp" to the query.

### 3.1 Implementation

We implemented the activity learning function as a Java application. Note that the search engine used in the experiment was MSN search because, unlike other search engines (e.g. Google search and Yahoo search engine), it has no access limitations (e.g. query limit of 1000 unique requests per user per day). In addition, Japanese was the language used in all experiments in this paper, however, our results can be applied to the English language if the lexico-syntactic pattern is tuned for the English language.

To navigate the activity-model learned from the web in an intuitive manner, we use Relation Browser[*1] developed by M. Stefaner. Relation Browser was designed to visualize conceptual structures, social networks, or anything else that can be expressed as a set of nodes and links. Figure 2 shows the current user interface for navigating the activity-model. At first, the user selects the domain, then the top-activity, and finally the sub-activity associated with the chosen top-activity.

### 3.2 Comparison of methods to estimate hierarchical relation

This section investigates which of methods 1-4 is the best for correctly associating sub-activity with top-activity, i.e. lowest error rate. The ground truth for this association was created by manually selecting both top-activities and sub-activities from the activities extracted from the web. Methods 1-4 were used to develop sub-activity and top-activity pairs as described in Section 2.2. We compare the error rates for methods 1-4. Error rate was calculated as follows:

$$
Errorrate = \frac{g_{+-} + g_{-+}}{g_{++} + g_{+-} + g_{-+} + g_{--}}
$$

where $g_{++}$, $g_{+-}$, and $g_{-+}$ are defined in Table 1. Here, function $prediction(p,c)$ outputs true when the mutual information of methods 1-4 between top-activity $p$ and sub-activity $c$ is the highest, and outputs false otherwise.

Table 2 list the sub-activity, correct top-activity (chosen by author), and top-activity estimated by methods 1-4. A shaded column indicates estimation failure, i.e. the wrong top-activity was associated with the sub-activity. For example, the activity "sell book" is chosen as the correct top-activity of the sub-activity "make book list". Therefore, Method 1 and 3 correctly estimated the top-activity, while methods 2 and 4 failed to do so.

In the book domain, there are three top-activity candidates: "sell book", "write book" and "read book". Both Method3 and Method4 had the lowest error-rate(0.222). As for the number of pairs correctly estimated, both Method3 and Method4 were best for the pairs that included "sell book" and "write book", while Method1 and Method2 were best for the pair that included "read book". However, both Method1 and Method2 estimated the top-activity of most sub-activities as "read book", and they had poor classification ability. This is shown in the error rate. Method1 and Method2 had higher error rates than Method3 and Method4 for the pair that had "read book" as top-activity. Therefore, we adopt Method3 as its average error rate is low and stable, unlike the other methods.

As for Method2, it had higher error rate than Method3. This indicates that, for correctly estimating the top-activity, dividing the top-activity into noun and verb parts is more effective than dividing the sub-activity. As we examine the combinations of one sub-activity with multiple top-activity candidates, extracting features from top-activity is important.

In the above experiment, we show the effectiveness of dividing the sub-activity to calculate distance between tasks,

表 1: Contingency table

| | label $y$=+1 | label $y$ =−1 |
|---|---|---|
| $prediction(p,c) = true$ | $g_{++}$ | $g_{+-}$ |
| $prediction(p,c) = false$ | $g_{-+}$ | $g_{--}$ |

[*1] http://moritz.stefaner.eu/projects/relation-browser/

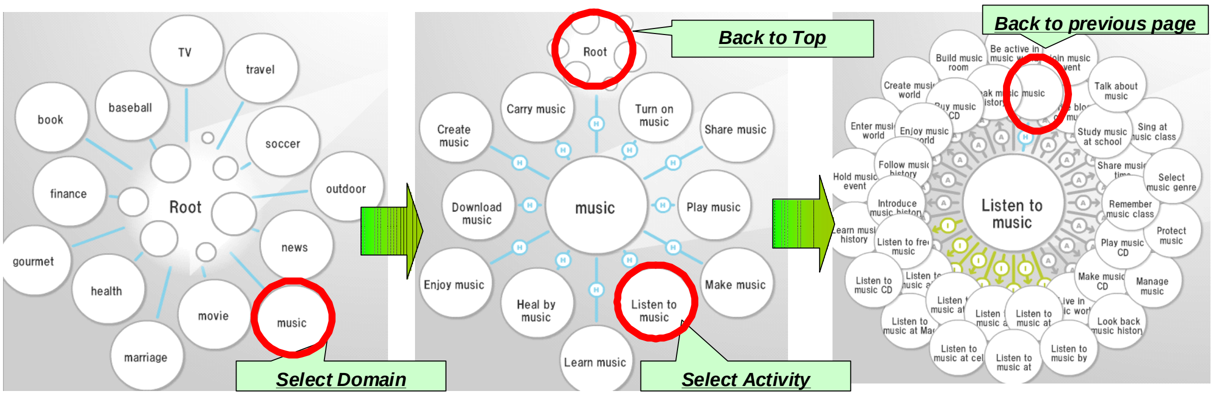

図 2: User Interface for navigating activity-model learned from the web

表 2: Comparison of error rate in estimating relationship between top-activity and sub-activity in the book domain.

| Sub-activity | Correct top-activity | Method 1 | Method 2 | Method 3 | Method 4 |
|---|---|---|---|---|---|
| Make book list | Sell book | sell | read | sell | write |
| Go out to book selling | Sell book | read | read | sell | sell |
| Open book store | Sell book | read | read | sell | sell |
| Manage book store | Sell book | read | read | sell | sell |
| Sell new book | Sell book | sell | read | write | write |
| Sell secondhand book | Sell book | sell | read | sell | read |
| Work for book publisher | Sell book | read | read | read | sell |
| Hold book recycling | Sell book | read | read | read | sell |
| Read foreign book | Read book | read | read | read | read |
| Teach book reading | Read book | read | read | read | read |
| Look for book author | Read book | sell | read | write | sell |
| Buy new book | Read book | read | read | read | read |
| Read book review | Read book | read | read | read | read |
| Write book review | Read book | read | read | write | write |
| Check book ranking | Read book | read | read | write | write |
| Check book recommendation | Read book | read | read | write | sell |
| Write book story | Write book | read | read | write | write |
| Write book novel | Write book | read | read | write | write |
| Write book manuscript | Write book | read | read | write | write |
| Finish up book manuscript | Write book | read | read | write | write |
| Write book essay | Write book | read | read | write | write |
| **# of correct estimation (sell book)** | 8 | 3 | 0 | 5 | 5 |
| **# of correct estimation (read book)** | 8 | 7 | 8 | 4 | 4 |
| **# of correct estimation (write book)** | 5 | 0 | 0 | 5 | 5 |
| **Total of correct estimation** | 21 | 10 | 8 | 14 | 14 |
| **Error rate (sell book)** | | 0.286 | 0.381 | 0.143 | 0.238 |
| **Error rate (read book)** | | 0.476 | 0.619 | 0.286 | 0.238 |
| **Error rate (write book)** | | 0.238 | 0.238 | 0.238 | 0.190 |
| **Average error rate** | | 0.333 | 0.413 | **0.222** | **0.222** |

however, the result is preliminary and is limited in terms of small number of top-activity (2 in book domain). In order to show the applicability of proposed method to other domains, more comprehensive study should be done.

## 4. Conclusion and Future Works

In this study, we investigated the automatic modeling of users' real world activities from the web. We conclude the paper by answering the research question: Is it possible to acquire hierarchical relationships between activities with low error rate? Yes, almost 80% of the hierarchical relationships could be captured by proposed method. The result, however, is limited in terms of small number of top-activity (2 in book domain). In order to show the applicability of proposed method to other domains, more comprehensive study should be done. In future works, we will create huge task-models that cover a wide variety of domains, and conduct user tests to evaluate the effectiveness of a task-based service navigation system.

## 参考文献

[1] K. Church, W. Gale, P. Hanks, and D. Hindle. Using Statistics in Lexical Analysis. In U. Zernik, editor, *Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon*, pages 115–164. 1991.

[2] K. Church and P. Hanks. Word association norms, mutual information and lexicography. In *Proc. COLING1989*, pages 76–83, 1989.

[3] Y. Fukazawa, T. Naganuma, K. Fujii, and S.Kurakake. Construction and use of role-ontology for task-based service navigation system. In *Proc. ISWC2006*, pages 806–819, 2006.

[4] Y. Fukazawa and J. Ota. Learning user's real world activity model from the web. In *Proc. Web Intelligence and Interaction*, 2009 (in Japanese).

[5] T. Naganuma and S. Kurakake. Task Knowledge Based Retrieval for Service Relevant to Mobile User's Activity. In *Proc. ISWC2005*, pages 959–973, 2005.

[6] M. Sabou, C. Wroe, C. Goble, and G. Mishne. Learning domain ontologies for web service descriptions: an experiment in bioinformatics. In *Proc. WWW2005*, pages 190–198, 2005.

[7] D. Sanchez. *Domain Ontology Learning from the Web.* VDM Verlag, 2008.