Automatic Modeling of User's Real World Activities from the Web for Semantic IR

Yusuke Fukazawa
NTT DOCOMO, Inc.
3-6, Hikari-no-oka, Yokosuka, Kanagawa, Japan
fukazawayuu@nttdocomo.co.jp

Jun Ota
The University of Tokyo
5-1-5 Kashiwanoha, Kashiwa, Chiba, Japan
ota@race.u-tokyo.ac.jp

ABSTRACT

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. The system allows the user to concretize his/her request in the task-model developed by human-experts. In this study, to reduce the cost of collecting a wide variety of activities, we investigate the automatic modeling of users' real world activities from the web. To extract the widest possible variety of activities with high precision and recall, we investigate the appropriate number of contents and resources to extract. Our results show that we do not need to examine the entire web, which is too time consuming; a limited number of search results (e.g. 900 from among 21,000,000 search results) from blog contents are needed. In addition, to estimate the hierarchical relationships present in the activity model with the lowest possible error rate, 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. The result shows almost 80% of the hierarchical relationships can be captured by the proposed method.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning - knowledge acquisition; I.2.7 [**Artificial Intelligence**]: Natural Language Processing - text analysis

General Terms

Algorithms, Performance, Design.

Keywords

Activity model, web mining, co-occurrence analysis

1. INTRODUCTION

The mobile Internet is expanding dramatically regardless of the metric used, 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[5][9] 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. Part of the task-model is shown in Fig.1. The connection between tasks is expressed by the *is-achieved-by* relation. The upper nodes of the task-model have generic tasks, while the lower nodes have more concrete tasks; the end nodes provide associations to services or contents via their URI. 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.

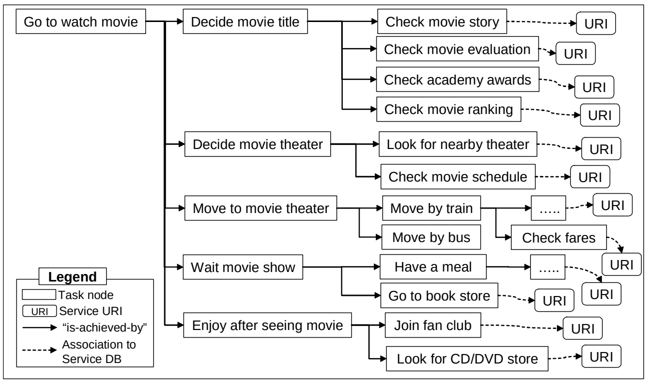


Figure 1: View of a part of task-model

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 and Twitter, 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

WWW2010, April 26-30, 2010, Raleigh, North Carolina.
.

into two types depending on what kinds of resources are used for learning, i.e. structured or unstructured data.

1.1.1 Model learning from structured data

This section describes past approaches to using structured data such as question-answering services, commonsense databases and know-how sites to create a comprehensive task-model.

- Learning from question-answering service

  Gupta and Kochenderfer [6] generated a task-model to make common sense data about indoor home and office environments through the Open Mind Indoor Common Sense (OMICS) website[1]. To convert the common sense of non-expert users into relations in a knowledge base, they use sentence templates. Users are prompted to fill in the blanks of sentences with words (e.g. they use the template "A computer is off when a ____ is ____." to capture causality).

  In order to motivate users to contribute common sense knowledge (OMICS), Lieberman et al. [7] proposed a question-answering game interface yielding a distribution of commonsense knowledge about a given topic with every iteration of the game.

- Learning from common sense knowledge database

  Shah and Gupta [17] proposed a method to construct a task-model as a first order Markov chain based on the set of order relationships between activities as present in the OMICS database.

  Pentney et al. [13][12][11] developed a system called SRCS, State Recognition using Common Sense, which reasons about the human state by creating statistical models of human activities. To build a model, SRCS gauges the degree to which individual relations in OMICS can be trusted using KnowItAll[2], and converts a weighted set of relations into Markov Random Fields(chain graph). KnowItAll is an information retrieval system designed to extract and evaluate widely known facts (and therefore also common sense) from the web.

  Mueller [8] proposed the automatic modeling of activities from narratives about restaurants. The model consists of a series of states (e.g. location, object) of characters (e.g. customers, a cook, and a waiter) at each time point, which are acquired by combining information extracted from narratives and commonsense axioms assembled by the authors.

- Learning from know-how site

  Perkowitz et al. [14] proposed to mine web sites like ehow.com for activity models. They extracted a set of objects involved in each step of an activity and trained Hidden Markov Models to match the sequences of RFID tag acquisitions.

  Nyga et al. [10] presented approaches to convert the natural language instructions for complex household chores given in wikihow.com into a logic-based representation and then to convert them into commands in the robot plan language RPL.

- Use of user generated data

  Many services have been commercialized, which motivate user to post their daily activity on the web such as Twitter or Blogs. To share the user input data in between multiple services, Chris Messina et al. prposed an extension to the Atom feed format to express what people are doing around the web.[3]

1.1.2 Model learning from unstructured data

- Learning from web

  Sabou et al. [15] 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 [16] 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.

- Learning from sensory data

  Eagle and Pentland proposed a reality-mining to identify the structure inherent in daily behavior with models based on cell phones to a fairly finite number of markove states by using longitudinal behavioral data[4]. They have shown that the method effectively extracts the underlying structure in the daily patterns of human behavior, predict subsequent behavior.

1.1.3 Summary of Related Works

Table 1 summarizes past approaches to the generation of task-models. Most of the past approaches, intended for structured data, were designed to model complex relationships (ordered relationships especially) in order to accomplish complex problem solving such as generating robot commands and estimating the user's behavior from sensor information. They did not focus on the acquisition of hierarchical relations between activities. Some approaches acquired Goal and Steps of two hierarchies, however, they did not estimate correct pairs of activities and sub-activity from many candidates of top-activities but instead used manually constructed pairs common in Q&A sites or know-how sites.

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, hierarchical relationships are more useful than order relationships. This is because when many activities exist in one hierarchy, navigation operation becomes complex, and it is often the case that the user will want to transition from an abstract desire to a more concrete desire. Therefore, in this paper, we propose a method that can automatically extract user activities from

[1] http://openmind.hri-us.com/

[2] http://www.cs.washington.edu/research/knowitall/

[3] http://activitystrea.ms/

Table 1: List of related works and comparison between related works and this paper

	target domain	resource	Kinds of relationship defined in activity-model	
			Order relationship	Hyponymy relationship
Gupta and Kochenderfer [5]	indoor home and office environments	question-answering through OMICS website	*extracted from QA and represented by Belief-Desires Intensions theory	* extracted from QA e.g. Question: "making a coffee requires the steps:____" Answer: Clean the coffee maker
Lieberman et al. [6]	everyday goals	question-answering through game interface	*extracted from QA pattern	* extracted from QA e.g. Question: "Why would you want to: *watch movie* ?" Answer: To have fun (parent goal of *watch movie*)
Pentney et al. [11]	indoor home and office environments	common sense data(OMICS) KnowItAll	*learned by Markov Analysis	-
Shah and Gupta [16]	indoor home and office environments	common sense data(OMICS)	*learned by Markov Analysis	* extraceted from plan (task and steps) defined in OMICS e.g. Task: wash clothes Steps: collect clothes, move to washing machine etc.
Mueller [7]	restaurant	common sense collected by auther	* extracted from common sense and represented by First-order logic	-
Perkowitz et al.[13]	household	ehow.com	* learned by Bayesian network	-
Nyga et al. [9]	cooking	wikihow.com	* extracted from ehow.com and represented by Reactive Plan Language	* extracted from goal and steps defined in wikihow.com e.g. Goal: How to set a table Step: Place placemat in front of chairs
Sanchez [15]	medical	web	-	-
Sabou et al. [14]	bioinformatics	web	-	-
this paper	**real world activities**	**web**	**-**	*** learned from the web using statistical analysis** **e.g. Activity: Watch movie** **Concrete activity: Buy movie ticket**

Web sources, and estimate the inclusive relations between activities for constructing a comprehensive task-model.

1.2 Research questions and approaches

To automatically extract the task-model desired from Web sources, we have the following two research questions.

1. How to extract activities with high precision (without noise) and recall (with enough coverage)

2. How to accurately acquire (i.e. low error rates) the hierarchical relationships between activities

Against research question 1, we adopt a method proposed by David [16]; it uses lexico-syntactic pattern analysis to extract task candidates and PMI based co-occurrence analysis to filter noise task candidates. David, however, did not discuss, which kind of site is appropriate and how many results were needed. Therefore, in this study, we investigate the most appropriate number of contents and resources to extract activities with high precision (without noise) and recall (with enough coverage).

Against research question 2, the problem can be considered as the calculation of semantic distance between two concepts. Therefore, 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.

2. EXTRACTING A TASK-MODEL FROM THE WEB

In this section, we propose a method to extract the widest possible variety of activities 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.2. 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.

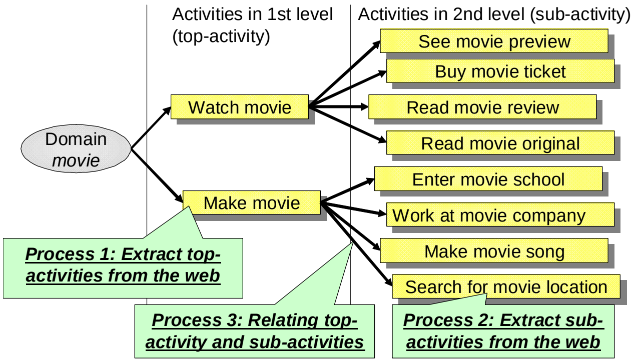

Activities in 1st level (top-activity)
Activities in 2nd level (sub-activity)
Domain
movie
Watch movie
Make movie
See movie preview
Buy movie ticket
Read movie review
Read movie original
Enter movie school
Work at movie company
Make movie song
Search for movie location
Process 1: Extract top-activities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

Figure 2: Structure of light-weight task-model

2.2 Defining domain concepts

We choose 31 content categories that are open to the public in Yahoo! Japan[4] from the viewpoint of supporting Japanese mobile users. Note that the activity extraction method described in the next section is applicable to not only the domains defined here but other domains defined in other ways.

2.3 Extracting activities in 1st level

[4] http://www.yahoo.co.jp

In order to extract activities in the 1st level (top-activity), which are strongly related to the domain, we collect the activities that consist of the domain as the noun part and verb that is strong related to the domain. To this end, we adopt the method proposed by David [16]; it uses lexico-syntactic pattern analysis to extract task candidates and PMI based co-occurrence analysis to filter noisy task candidates.

The first step is to extract activity candidates for the domain from the source texts. Concretely, we use the domain to form queries for a web search engine. The result is a set of web resources that contain context words. The web content is parsed and, using the lexico-syntactic pattern "VP (domain)", verb phrase candidates for the domain are obtained. Here, we acquire all activity candidates by joining the verb phrase and domain. Note that lexico-syntactic pattern used here is for English, and should be tuned for the language intended. In Japanese, the lexico-syntactic pattern would be " (domain) $PRE\ VB$".

The second step is to figure out, from among the activity candidates listed, which are correct and are most strongly related to the domain by using a measure based on co-occurrence analysis. Turney proposed PMI (Pointwise mutual information)[2][1] based co-occurrence analysis that uses a web search engine data[18]. In [18], he considered the synonym test question, one of 80 TOEFL questions. Let *problem* represent the problem word and choice 1, choice 2, ..., choice n represent the alternatives. The score, the measure of co-occurrence between $problem$(problem word) and $choice$(related candidate concepts), is given as follows:

$$Score(choice, problem) = \frac{hits(\ problem\,AND\,choice\ )}{hits(problem)\dot{h}its(choice)}$$

where $hits$(key) means the number of results the search engine returns against the query "key". This co-occurrence measure has been extensively used to evaluate the relevance of a set of candidates[3][16]. Here, we rewrite it as

$$Score(domain, verb) = \frac{hits(\ \text{“}domain\text{”}\ AND\ \text{“}verb\text{”}\ )}{hits(\text{“}verb\text{”})}$$

.

Note that we eliminate $hits$("$domain$") from the denominator as this value is common in all candidates. Once scores for all candidates have been computed, those that exceed a threshold are selected. We empirically set 10^{-5} as the threshold.

Table 2: Activity extraction process

Web Query	"movie"
URL	http://www.bing.com/search?q="movie"
Sample Text	...where the rumor in hollywood is where he is going to **film** movie....Keep up to date with the latest releases and **watch** movie trailers online at...
Candidate evaluation	Hits("film" AND "movie")=3,030,000 Hits("film")= 12,000,000 Score= **0.25**
	Hits("watch" AND "movie")=1,230,000 Hits("watch")=1,430,000 Score= **0.86**

2.4 Extracting activities in 2nd level

2.4.1 Definition of 2nd level activity

We define 2nd level activities (sub-activities) as activities that represent specializations of a top-activity. Here, we consider specialization of the noun part and of the verb part of the top-activity independently. In general, we cannot obtain specialization of verbs without annotating the noun phrase to the verb(e.g. specialization of "watch" is obtained by annotating a noun such as "movie" which yields "watch movie"). Therefore, we consider specialization of the noun part to obtain specialization of the top-activity. To automatically extract noun part specialization, we adopt the pattern-based approach for detecting object specialization through noun phrases[16]. In the English language, the immediate anterior word for a keyword frequently expresses a semantic specialization (e.g. *credit card*). We can use the same pattern-based method as the method to extract sub-activity by finding noun phrase that includes domain and verbs related to the noun phrase.

2.4.2 Steps to extract sub-activities

The first step is to extract noun phrases that includes domain from the web. Concretely, we use the domain to form queries for a web search engine. The web content is parsed and, using the lexico-syntactic pattern "NP (domain)" and "(domain) NP ", noun phrase candidates are obtained. Here, we acquire all noun phrase candidates by joining the obtained noun phrases and domain. Note that the lexico-syntactic pattern used here is for English, and should be tuned for the language intended.

The next step is to extract sub-activity candidates for each noun phrase so obtained. We take the same approach as described in Section 2.3 for each obtained noun phrase;, the process is 1)extract verb candidates using the lexico-syntactic pattern "VP (obtained noun phrase)", and 2) use co-occurrence analysis to choose verbs that are most strongly related to the noun phrase.

2.5 Relating sub-activity to top-activity

This step figures out, from among the sub-activity candidates listed, 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 (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.

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. $pmi(x, y) = \log\left(\frac{hits(x\text{AND}y)}{hits(x)hits(y)}\right)$. We call this approach Method1. The score calculation described in Section 2.3 is derived from the above equation.

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:

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

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

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

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

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:

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

Here, we set $pmi(p_n, c_n)$, $pmi(p_n, c_v)$ and $pmi(p_n, c_n \text{AND} c_v)$ 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 answer two research questions posed in Section 1.2. To answer question 1, we evaluate two things: quantity of search results needed to acquire enough top-activities in Section 3.2, and kinds of sites appropriate for acquiring activities with high precision and recall in Section 3.3. To answer question 2, 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 3.4. The kinds of site compared in this experiment include the entire Web, blogs[5], Twitter[6], Q&A sites[7], and news[8]. 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 as described in Sections 2.3 and 2.4.2.

[5] http://blogs.yahoo.co.jp

[6] http://twitter.com

[7] http://questionbox.jp.msn.com

[8] http://topics.jp.msn.com

3.2 Number of search results needed

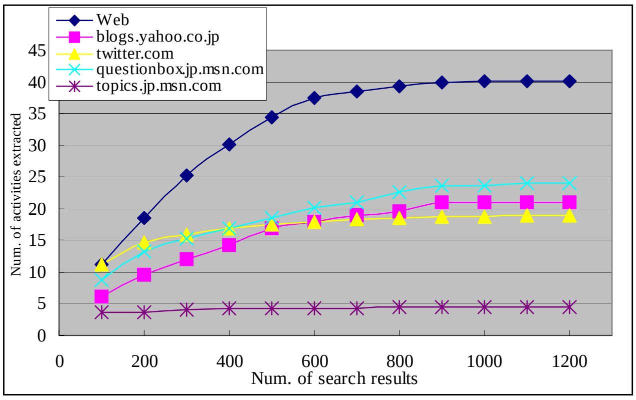


Figure 3: Comparison of number of activities extracted versus the number of search results.

In this section, we investigate the number of contents needed to acquire sufficient activities. Considering that there is a limit to the number of actions that the user can realistically take in any specific domain, we assume that if a certain number of contents are examined, we can cover most of the activities the user can take. Accordingly, we investigate the number of activities acquired against the number of retrieval results.

In Figure 3, the average number of top-activities extracted in the 31 domains is plotted against the number of retrieval results. For all types of sites, the number of activities extracted initially increases linearly. The number of extracted activities begins to saturate at around 900 retrievals. The minimum number of search result depends on the domain; however, we can say that even a limited number of search results is enough to cover a wide variety of activities. Note that analyzing 900 contents and extracting the activities took almost 30 minutes on a mid-range PC, which is fast enough in practice. Analyzing overall search results (e.g. 21,000,000 search results for movie domain) would take about 12,000 hours (=500 days), which is slower than constructing the task-model manually.

Next, we compare the number of activities extracted from each site for 900 retrieval results. An average of 40.0 activities can be extracted from the entire Web. Q&A site (23.6 pieces), Yahoo blog (21.0 pieces), Twitter (18.3 pieces) and news contents (4.4 pieces) follow the entire web. As the users had different purposes when accessing different sites, different kinds of activities can be extracted. As users described opinions or problems that occurred in daily life in user-generated content such as blogs, Q&A sites, and Twitter, they cover almost the same range of activities. News sites offer a common topic to a lot of users; just a few activities are used often. As the entire Web includes all sites, meaningless activities (noise) are likely to be acquired along with the many kinds of activities. We show evidence of this in the next section.

3.3 Kind of site appropriate for activity acquisition

In this section, we compare the precision and recall of the activities extracted from each site type. We choose the book domain and movies because they yielded the greatest number of activities in the 900 search results acquired in the experiment described in the previous section. Table 4 and Table 5 compare the precision and recall values of activities learned from each kind of site for the movie domain and the book domain. The integers in the table are the activity rank on co-occurrence score. An empty column means that the corresponding activity could not be extracted from that site type. Note that only correctly extracted activities are listed in the figure. To collect the ground truth of top-activity, authors excluded meaningless activities, which are impossible to perform in the real world. For example, “watch movie” was extracted with the highest co-occurrence score from the entire web, however, the list has no second or third rank entries. This is because both second rank and third rank activities were judged as meaningless. Precision, recall and F-measure values are also shown in the tables for 5 and 20 top activities. Precision, recall and F-measure values were determined as follows:

$$precision = \frac{f_{++}}{f_{++} + f_{+-}}, recall = \frac{f_{++}}{f_{++} + f_{-+}}$$

$$F - measure = \frac{2 \cdot precision \cdot recall}{precision + recall}$$

where f_{++}, f_{+-}, and f_{-+} are defined in Table 3.

In the movie domain, as can be seen from Table 4, for both the top 5 and top 10, f-measure of the entire web is very low while that of blogs is the highest. In the book domain, as can be seen from Table 5 , for both the top 5 and top 10, f-measure of blogs is also the highest. As discussed in the previous section, the entire Web covers such a wide variety of activities that meaningless activities (noise) are likely to be acquired. Compared to the other user-generated content (Q&A or Twitter), blog users described what he/she experienced or wanted to experience in daily life in detail, so many meaningful activities can be found in these sites. That is, blogs contain a wide variety of activities with low noise. Therefore, these results show that blogs are the best resource from which to extract real world user activities.

3.4 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 2.4. 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 6. Here, function $prediction(p, c)$ outputs true when the mutual informa-

Table 3: Contingency table

	label y=+1	label y =−1
rank > $TopN$	f_{++}	f_{+-}
rank ≤ $TopN$	f_{-+}	f_{--}

Table 4: Comparison of precision and recall of top-activity in the movie domain.

Correct Activity (label y=+1)	**web**	**blog**	**twitter**	**QA**	**News**
Enjoy movie	11	9	-	-	5
Make movie	10	-	3	12	4
Watch movie	1	2	1	2	3
Act in movie	12	8	-	11	6
See movie	2	1	11	1	1
Go to watch movie	16	12	14	3	-
Remember movie	15	3	16	-	-
Cry at movie	14	4	15	-	-
Invite to movie	19	5	13	4	-
Go to movie	22	-	2	7	-
Look for movie	30	-	-	6	-
Film movie	18	11	5	15	2
Precision(Top5)	0.400	**1.000**	0.800	0.800	**1.000**
Recall(Top5)	0.154	**0.385**	0.308	0.308	**0.385**
F-measure(Top5)	0.222	**0.556**	0.444	0.444	**0.556**
Precision(Top10)	0.300	**0.700**	0.400	0.600	0.600
Recall(Top10)	0.231	**0.538**	0.308	0.462	0.462
F-measure(Top10)	0.261	**0.609**	0.348	0.522	0.522

Table 5: Comparison of precision and recall of top-activity extracted in the book domain.

Correct activity (label y=+1)	**web**	**blog**	**twitter**	**QA**	**news**
Write book	16	12	6	3	2
Publish book	7	-	5	19	3
Read book	4	14	-	-	1
Recommend book	24	-	-	-	-
Search for book	13	7	-	9	-
Make book	8	1	7	2	-
Buy book	6	5	2	16	-
Sell book	17	3	3	11	-
Borrow book	18	2	-	7	-
Select book	21	6	-	-	-
Record in the book	23	4	4	14	-
Meet the book	22	-	-	5	-
Present book	20	-	-	18	-
Precision(Top5)	0.200	**1.000**	0.800	0.600	0.600
Recall(Top5)	0.077	**0.385**	0.308	0.231	0.231
F-measure(Top5)	0.111	**0.556**	0.444	0.333	0.333
Precision(Top10)	0.400	**0.700**	0.600	0.500	0.300
Recall(Top10)	0.308	**0.538**	0.462	0.385	0.231
F-measure(Top10)	0.348	**0.609**	0.522	0.435	0.261

tion of methods 1-4 between top-activity p and sub-activity c is the highest, and outputs false otherwise.

Tables 7 and 8 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 "make movie" is chosen as the correct top-activity of the sub-activity "write movie original". Therefore, Method 1 correctly estimated the top-activity, while methods 2-4 failed to do so.

In the movie domain, there are two top-activity candidates: "watch movie" and "make movie". All pairs that had "watch movie" as top-activity were estimated correctly by all methods. On the other hand, Method 3 was best in estimating the pair that had "make movie" as top-activity, and had the lowest error-rate(0.2). This means that 80% of the hierarchical relationships could be captured by Method3.

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 in both movie and book domains. 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.

As for Method4, its average error rate equals that of Method3 (0.222) in the book domain; however, Method4 had the highest error rate in the movie domain. We assume that if there are more than 3 top-activity candidates or there is a bias in the frequency of top-activities, we should extract more features from top-activity and sub-activity to estimate top-activity correctly.

In the above experiment, we show the effectiveness of dividing the sub-activity to calculate distance between tasks, however, the result is preliminary and is limited in terms of small number of top-activity (2 in book domain and 3 in movie domain). In order to show the applicability of proposed method to other domains, more comprehensive study should be done.

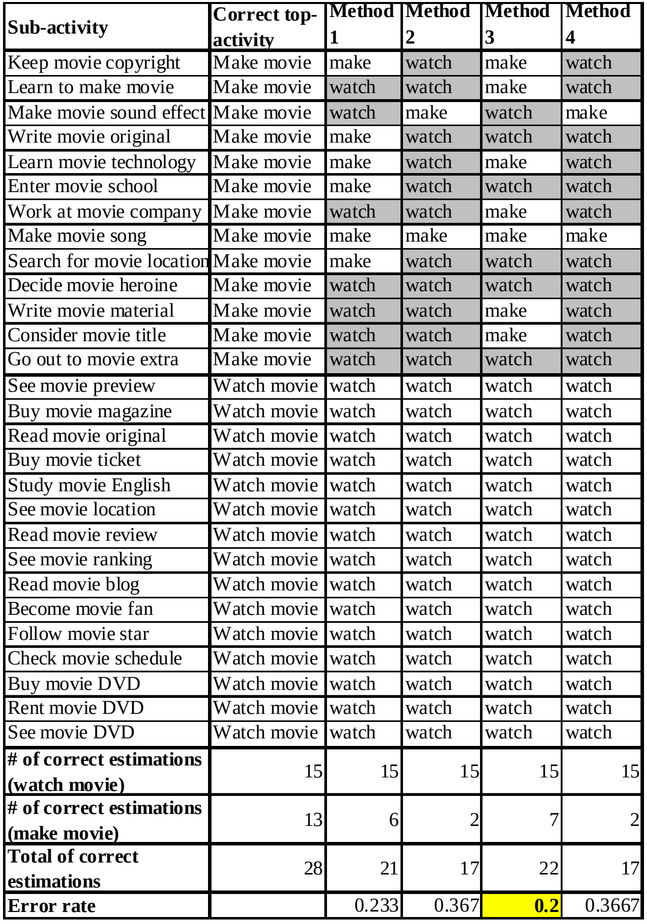
4. CONCLUSION AND FUTURE WORKS

In this study, we investigated the automatic modeling of users' real world activities from the web. We conclude the paper by answering the research questions in Section 1: 1) Is it possible to extract activities with high precision (without noise) and recall (with enough coverage)? Yes. We found that we do not need to examine all web contents, which would be excessively time consuming; just a limited number of search results (e.g. 900 from among 21,000,000 search results) are needed to extract activities with high precision and recall. This took only 30 minutes, which is fast enough in practice. Next question: 2)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

Table 6: Contingency table

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

Table 7: Comparison of error rate in estimating relationship between top-activity and sub-activity in the movie domain.



Sub-activity	Correct top-activity	Method 1	Method 2	Method 3	Method 4
Keep movie copyright	Make movie	make	watch	make	watch
Learn to make movie	Make movie	watch	watch	make	watch
Make movie sound effect	Make movie	watch	make	watch	make
Write movie original	Make movie	make	watch	watch	watch
Learn movie technology	Make movie	make	watch	make	watch
Enter movie school	Make movie	make	watch	watch	watch
Work at movie company	Make movie	watch	watch	make	watch
Make movie song	Make movie	make	make	make	make
Search for movie location	Make movie	make	watch	watch	watch
Decide movie heroine	Make movie	watch	watch	watch	watch
Write movie material	Make movie	watch	watch	make	watch
Consider movie title	Make movie	watch	watch	make	watch
Go out to movie extra	Make movie	watch	watch	watch	watch
See movie preview	Watch movie	watch	watch	watch	watch
Buy movie magazine	Watch movie	watch	watch	watch	watch
Read movie original	Watch movie	watch	watch	watch	watch
Buy movie ticket	Watch movie	watch	watch	watch	watch
Study movie English	Watch movie	watch	watch	watch	watch
See movie location	Watch movie	watch	watch	watch	watch
Read movie review	Watch movie	watch	watch	watch	watch
See movie ranking	Watch movie	watch	watch	watch	watch
Read movie blog	Watch movie	watch	watch	watch	watch
Become movie fan	Watch movie	watch	watch	watch	watch
Follow movie star	Watch movie	watch	watch	watch	watch
Check movie schedule	Watch movie	watch	watch	watch	watch
Buy movie DVD	Watch movie	watch	watch	watch	watch
Rent movie DVD	Watch movie	watch	watch	watch	watch
See movie DVD	Watch movie	watch	watch	watch	watch
# of correct estimations (watch movie)	15	15	15	15	15
# of correct estimations (make movie)	13	6	2	7	2
Total of correct estimations	28	21	17	22	17
Error rate		0.233	0.367	0.2	0.3667

in terms of small number of top-activity (2 in book domain and 3 in movie 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.

5. ACKNOWLEDGEMENT

The authors thank Prof. Yutaka Matsuo for his helpful comments on early research plan.

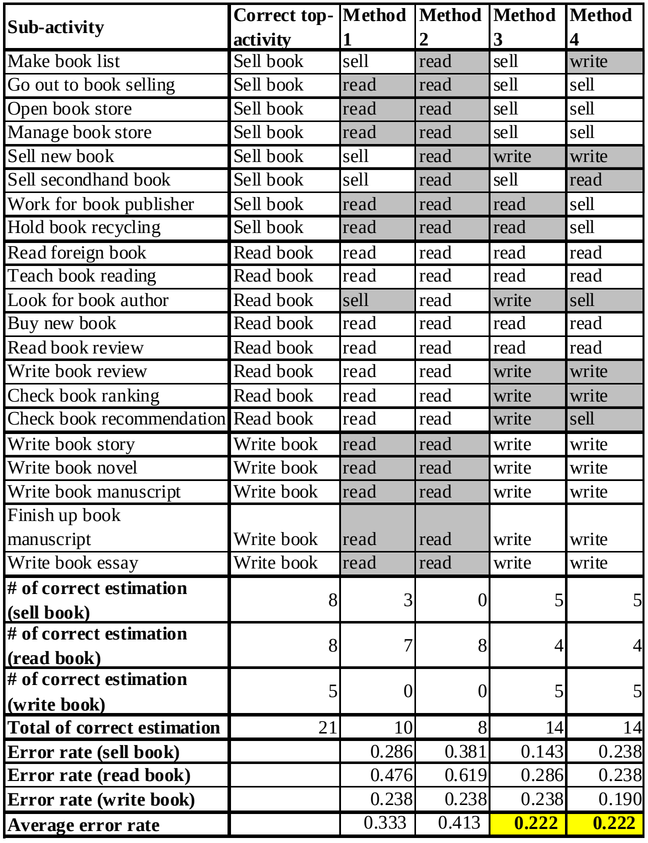
6. REFERENCES

[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] P. Cimiano and S. Staab. Learning by Googling. *SIGKDD Explorations Newsletter*, 6(2):24–33, 2004.

Table 8: 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

[4] N. Eagle and A. Pentland. Eigenbehaviors: Identifying structure in routine. *Behavioral Ecology and Sociobiology*, 7(63):1057–1066, 2009.

[5] 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.

[6] R. Gupta and M. J. Kochenderfer. Common sense data acquisition for indoor mobile robots. In *Proc. AAAI2004*, pages 605–610, 2004.

[7] H. Lieberman, D. Smith, and A. Teeters. Common Consensus: a web-based game for collecting commonsense goals. In *Proc. of the Workshop on Common Sense and Intelligent User Interfaces at IUI2007*, 2007.

[8] E. T. Mueller. Modelling space and time in narratives about restaurants. *Literary and Linguistic Computing*, 22(1):67–84, 2007.

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

[10] D. Nyga, M. Tenorth, and M. Beetz. Understanding and executing instructions for everyday manipulation tasks from the world wide web. Technical report, IAS group, 2009.

[11] W. Pentney, M. Philipose, and J. A. Bilmes. Structure

learning on large scale common sense statistical models of human state. In *Proc. AAAI2008*, pages 1389–1395, 2008.

[12] W. Pentney, M. Philipose, J. A. Bilmes, and H. A. Kautz. Learning large scale common sense models of everyday life. In *Proc. AAAI2007*, pages 465–470, 2007.

[13] W. Pentney, A.-M. Popescu, S. Wang, H. A. Kautz, and M. Philipose. Sensor-based understanding of daily life via large-scale use of common sense. pages 906–912.

[14] M. Perkowitz, M. Philipose, K. Fishkin, and D. J. Patterson. Mining models of human activities from the web. In *Proc. WWW2004*, pages 573–582, 2004.

[15] 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.

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

[17] C. Shah and R. Gupta. Building Plans for Household Tasks from Distributed Knowledge. In *Workshop on Modeling Natural Action Selection at IJCAI*, pages 539–545, 2005.

[18] P. Turney. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *Proc. of the European Conference on Machine Learning*, pages 491–502, 2001.