December 2005

A Study of Context Inference Algorithm on the Web-based Information Systems

Sangjun Kim
University of Science and Technology, Rep. of Korea

Euiho Suh
University of Science and Technology, Rep. of Korea

Keedong Yoo
University of Science and Technology, Rep. of Korea

A Study of Context inference algorithm on the web-based information systems

Sangjun Kim
Dept. of IME, Pohang University of Science and Technology, Rep. of Korea
nataiji@postech.ac.kr

Euiho Suh
Dept. of IME, Pohang University of Science and Technology, Rep. of Korea
ehsuh@postech.ac.kr

Keedong Yoo
Dept. of IME, Pohang University of Science and Technology, Rep. of Korea
kdy@postech.ac.kr

Abstract

Recently, context-awareness has been a hot topic in the ubiquitous computing field. Numerous methods for capturing, representing and inferring context have been developed and relevant projects have been performed. Existing research has tried to determine user's contextual information physically by using stereo type cameras, RFID, smart devices, etc. Especially, these are focusing on physical context such as location, temperature, light, etc. However, cognitive elements of the context are important and needs more study.

Therefore, this paper confines its research domain to the web-based information system (IS) and proposes an algorithm for inferring cognitive context in the IS domain. First, a context inference algorithm aims at recognition of a user intention on the IS. To apply this algorithm, we use a text categorization technique to classify various text-based sources users are using such as web pages, pdf document, MS-word documents, etc. Second, the service inference algorithm is based on the similarity measurement between the user preference and the deliver-enabled services and tries to recommend a user-adaptive service to the user.

The context and service inference algorithm that this paper proposes can help the IS user to work with a IS conveniently and enable an existing IS to deliver ubiquitous service. In this fashion, the paper shows the direction an existing IS heads and ultimately, shows the typical services of a ubiquitous computing-based IS.

Keywords: Cognitive context, Text categorization, Context inference, Service inference, Ubiquitous computing

1. INTRODUCTION

Context-awareness is a core concept in the vision of ubiquitous computing. Context is defined as any information that can be used to characterize the situation of an entity (Prekop & Burnett 2003). Schilit et al. (1994) classified context into three categories: user context (user profile, location, people nearby, activity, etc.), execution context (network traffic, status of the device, communication costs, etc.) and environment context (weather, light, noise level, temperature, time, etc). Being aware of the context allows individuals to interact with systems that are aware of the environmental and computational state of an individual (Schilit et al. 1994).

There are a certain number of studies on context-awareness and inference. For example, Microsoft's 'EasyLiving' project has been designed to collect contextual information and deliver its applications (Brumitt & Meyers 2000). Within the 'EasyLiving' environment 'disaggregated computing' allows for a user's location and preferences to determine which set of inputs and outputs, across a set of computers, were connected to the currently active applications (West et al. 2004). Xerox PARC's PARCTAB is a Personal Digital Assistant (PDA) that communicates with a user via infrared data packets to a network of infrared transceivers (Schilit 1993). Using it, a user's location can be traced. Additionally, Hewlett Packard's 'Cool town' project extends the idea of context based on physical location by marking up the physical space with a mobile WWW infrastructure (Prekop & Burnett 2003). These projects try to be aware of physical elements such as location, light, sound, etc. by using stereo type cameras, RFID, smart devices, etc., and in this manner, consider a user's physical context. Currently, most attempts to use context-awareness within ubiquitous computing environments are centered on the physical elements of the environment, a user, or devices (Prekop & Burnett 2003). However, these studies have a limitation related to recognizing user intention in a static condition. In other words, it's difficult to recognize user's cognition context, not a physical context. For example, if a user browses web pages on the internet, it's difficult to detect the user's intention in the state of web browsing by using physical devices. Therefore, many authors acknowledged the importance of capturing the cognitive elements of a user's context (Schilit et al. 1994).

For awareness and inference of a user's cognitive context, we have to apply other methods that enable context inference. Hence, the suggestion of an alternative method is the ultimate goal of this research. In this manner, the paper confines the research domain to the web-based information system (IS) and suggests an algorithm for inferring user's cognitive context and delivering personalized service to the IS user.

In this study, we regard an intention of user as a cognitive context when user's using a web-based IS. To recognize a user's intention, we should firstly discern the sources that user's using on the IS and extract the representatives of each source. This can be accomplished by using a text categorization technique. Various sources used by user reflect the user's intention as a cognitive context. Consequently, we can infer the user's context by considering the combination of each source category synthetically. Perception of user's intention enables the system to recommend the personalized service to the user. System can obtain a service list that adaptive to the user's intention, and infer user-oriented services in the service list considering user preferences. Therefore this paper suggests,

- a context inference algorithm for recognizing a user's cognitive context on the web-based information system.
- a service inference algorithm for selecting user-adaptive services

The rest of this paper is organized as follows. Section 2 reviews existing research on context aware application and context inference. In section 3, we show the prerequisite for a context and a service inference algorithm. Next, we suggest and explain the algorithm, and a brief case study is shown in Sections 4 and 5, respectively. Lastly, section 6 provides our conclusions and directions for future work.

2. LITERATURE REVIEW

2.1 Existing context aware applications

A context aware application is defined as one that uses the context of an entity to modify its behavior to best meet the context of the user (Abowd et al. 1999). Schilit et al. (1994) classifies context aware applications into four cases based on the method of context inference. The first is the proximate selection which is a user-interface technique where objects located nearby are emphasized or easily selected for usage. Second is the automatic contextual reconfiguration. A reconfiguration process that adds new components removes existing components or alters their communication in accordance to context changes. Problems can arise if the context is changing rapidly and adaptation is triggered for every change or if the context is incorrectly reported and leads to unnecessary reconfiguration. Third is contextual information and commands, based on the assumption that people's actions can sometimes be predicted by studying their situational context. Queries on contextual information may produce different results according to the context in which they are issued. Likewise, commands may exploit such awareness to provide more efficient services. Fourth is the context-triggered action. Simple IF-THEN rules are used to specify how context-aware systems should behave and eventually adapt when a certain type of context occurs. In this time, the context in which an action is required to be triggered may consist of many conditions, meaning that it may be based on compound contexts or single context atoms. This paper pursues this type of context aware application.

Most of the context-aware applications described so far tend to focus on the external dimension of context, the elements of the physical environment, including location, proximity to other objects, temperature, time, lighting levels, and so on (Gwizdka 2000). To extend context-aware application into more cognitive domains, such as information retrieval, decision making, situation monitoring, product design, and so on, the internal dimension of context - user's goals, tasks, work context, business processes, personal events, communication, emotional and physical state - also needs to be captured (Gwizdka 2000).

2.2 Context inference

Multiple sensors are used to infer various contexts' complex features. For instance, light, sound, and temperature information permit the drawing of a conclusion about high-level context such as 'indoor' or 'outdoor'. In other words, an atom context such as light, sound, temperature, etc. is transformed into much higher level contexts. This context transformation process is called context inference.

West et al. (2004) presented support planes from ubiquitous system architecture called 'Nightingale'. In 'Nightingale', context plane was composed to simplify the integration of new users and devices in their pervasive computing environment, and to allow their applications to be context-aware. The context plane used the combination of user-driven, rule based, probabilistic and temporal logic inference techniques to carry out multiple context inference. Ranganathan & Campbell (2003) proposed a middleware that facilitated the development of context-awareness agents. In their paper, for inferring new contexts from existing one, they used static rules to deduce higher level context and machine learning techniques. These two studies are classified into the context-triggered action of all types described by Schilit et al. (1994).

Rule-based methods that are used for inferring user's context have a limitation that it needs a lot of rules when describing complicated environments. However, in rule-based methods, individual rules are easy to construct. Besides, rule-based methods are used easily for context inference in the case that the number of higher level context is limited. On domain of this paper, web-based information system, there is a small number of higher level contexts and it's easy to describe. Therefore the rule-based method is adapted to this paper.

3. PREREQUSITE FOR INFERENCE

The first step for inferring user's context on the web-based IS is to categorizing each source that user is using. This process is based on the text categorization techniques. Text categorization, one of the areas in text mining, is the method that automatically sorts text-based documents into predefined categories. For example, system assigns themes such as sports, finance, or politics to the categories of general interest. The standard modern approach to document categorization is machine learning method (Sebastiani 2002). There are various

kinds of methods based on machine learning that can be divided into 5 methods: manual categorization, clustering approach, META tags based categorization, link and content analysis, and text content based categorization (Asirvatham & Ravi 2001). Among them, text content based categorization is used widely for document categorization, capable of rendering good results in a way that is robust and that makes few assumptions about the context to be analyzed (Pierre 2001). Therefore, in this paper, we use the text-content based method for text categorization.
To perform document categorization based on machine learning, a corpus must be constructed in advance. The corpus stands for the large amount of pre-categorized data set. The corpus must be trained before entering new sample and training process is as follows.

• Tokenizing
It is necessary to break the text into discrete units, each usually corresponding to a word in the text. For example, articles and conjunctions are used frequently in sentences, but they are useless in text categorization process. Therefore they must be eliminated for the precise categorization
• Word stemming
It is necessary to standardize word's suffixes. For example, 'go' is a more appropriate word than 'going' for analyzing the text.
• Feature selection and weighting
When a common text is represented in a vector form, a dimensionality problem arises. To solve this problem, significant features in the text must be selected. The general algorithm is TFIDF (Term Frequency and Inverse Document Frequency). In this algorithm, terms may be weighted by their frequency in the document. Term weight may also be reduced by a factor representing the term's prevalence in other documents (Yang et al. 2002). The following formula (1) shows the TFIDF algorithm (Yang et al. 2002).

$$w_d(t) = (1 + \log_2 N(t,d)) \times \log_2(|D| / n(t)) \tag{1}$$

Where,
D is a set of training documents
n(t) is the number of training documents containing *t*
N(t, d) is the number of occurrence of *t*

Source formats such as PDF, MS-word, text, etc. that user is using are various. Therefore, for applying various source formats to categorization process, an additional process should be supplemented. For example, if the source type is a webpage, a meaningless HTML tag must be eliminated. Therefore, in this case, we should add a HTML tag elimination process in the categorization process.
Constructing the corpus is very important, because the quality of the corpus has critical influence on the result of the algorithm. A general corpus used in existing research such as the TREC, Reaters-22589 and OSHUMED is too vast and the criteria of categorization is not adaptable to this paper's domain. Therefore, for precise categorization, corpus that is adapted to paper's domain should be constructed.

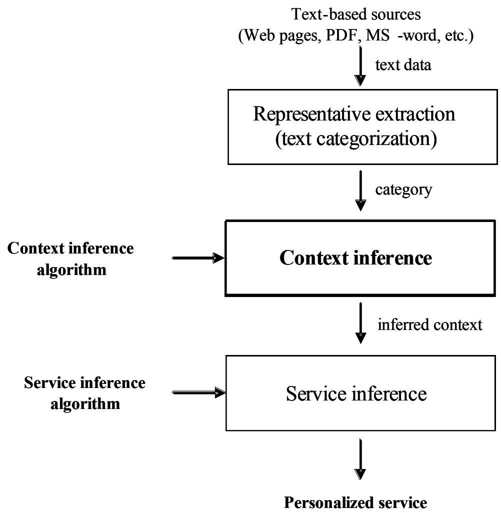
4. CONTEXT AND SERVICE INFERENCE FRAMEWORK

When a user perceives a problem and tries to find a solution in the IS, the user refers to the related information in form of WebPages, MS word files, etc. We can say that these sources referred to by the user reflect the user's context, excluding such extreme situations as watching a movie, drawing a picture, or listening to music. Therefore, by analyzing various sources the user resorts to, we can infer the user context.

Beforementioned sources are based on the text. In the case of WebPages, we can extract the text by analyzing HTML codes. The extracted text is used for determining representatives of each source. And to accomplish this, text categorization technique is applied. The text is processed by the text categorization procedures. Therefore, we can obtain a category as a representative of each source. The obtained categories are matched with the pre-defined categories in IS context memory, which contains various IS context. The IS context is the user's intention or user's cognitive context that we try to infer. And it can be obtained in advance following the IS structure and contents analysis. The pre-defined categories are assigned to each IS context. These categories are matched with the categories which the text categorization process results in. By comparing and scoring process between the categories, IS context is determined.

By using the inferred context and the user preference recognized in advance, we can make a deliver-enabled service list. A service inference algorithm selects user-suitable services from the list to recommend a personalized service within the IS.

Figure 1 shows the context and service inference framework.



<Figure 1> Context and service inference framework

4.1 Representative extraction

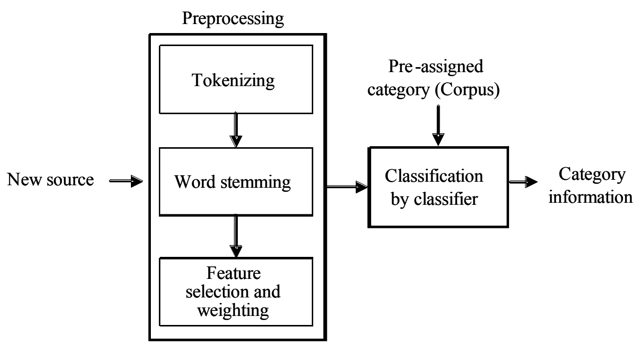
As mentioned above, representative extraction is based on the text categorization method. After establishing a pre-assigned category training set (corpus), an appropriate classifier categorizes a new webpage into a pre-determined category. In this procedure, a new sample and a corpus must be preprocessed following the methods shown in Figure 2, which include tokenizing, stemming, feature selection and weighting, closely explained in Section 3. And additional processes can be supplemented in the same manner in accordance with source types.

The actual categorization process is performed by classifiers which categorize a new sample. There are many classifiers for document categorization; among them Naïve Bayes, probabilistic classifiers, decision tree classifiers, decision rules, regression methods, neural network, the *k*-NN classifier, the support vector machine Rocchio classifiers are mostly used ones. The *k*-NN (*k*-Nearest Neighbor) classifier, one of the top performing method, is an approach effective for a broad range of pattern recognition and text classification problems (Yang & Liu 1999). In the *k*-NN algorithm, when given a document, the system finds the *k* nearest neighbors among the training set, and uses the categories of the k neighbors to weigh the category candidates. The similarity score of each neighboring document to the entering document is used as the weight of the categories of the former document. By sorting the scores of candidate categories, a ranked list is obtained for the new document. Using these scores as a threshold, binary category assignments are obtained (Yang & Liu 1999). Formula (2) shows the similarity measured by using cosine value (Ko et al. 2004).

$$s(c_i, d) = \sum_{d' \in R_k(d) \cap D_i} \cos(d', d) \qquad (2)$$

where,
d is a test document
D_i is a set of training document
C_i is a candidate category
$R_k(d)$ is the *k* nearest neighbors (training set) of document *d*

Figure 2 shows the procedure of representative extraction



< Figure 2 > Extracting representative procedure

4.2 Context inference algorithm

Text category data are used to infer user intention through the context inference algorithm. For this purpose, IS context must be analyzed in advance and stored in the memory. After perceiving IS context, the pre-defined categories are assigned to each IS context. Figure 3 shows the context/category information stored in IS context memory. Text category data are matched with the pre-defined categories in the IS context memory.
Prior to the main inference process, we need to state a few assumptions and considerations. First of all, there are a few assumptions.

• We neglect the extreme situations such as watching a movie or listening to music.
• It is impossible to infer user context by analyzing only one source. Therefore, algorithm is launched when more than 2 sources are existed.
• This paper ignores such exceptional cases as a mismatch between an inferred category and a pre-defined category in the IS context memory.

And we see some considerations for deploying algorithm

• The existing window screen that the user is browsing determines user context rather than the preceding or later window screens.
• Each context has a critical category which influences user context.
• The upper level category is weighted more than the lower level category to complement webpage categorization algorithm errors.

Based on these assumptions and considerations, we can launch the context inference algorithm.

Step 1. The first window screen category is determined. The context that includes this category in IS context memory is activated.
Step 2. The second window screen category is determined. The context that includes this category in IS context memory is activated as in step 1. The total score is calculated according to formula (3) and the context with the maximum total score is selected.
Step 3. If the user acknowledges this context positively, the algorithm function is terminated and system determines the selected context as a user's cognitive context. Otherwise, repeat step 2.

$$
\begin{array}{cc}
\textbf{IS Context} & \textbf{Pre-assigned category} \\
\end{array}
$$
$$
\left(\begin{array}{c|ccccc}
c_{11} & a_{21} & a_{31} & a_{41} & \vdots & a_{n1} \\
c_{12} & a_{22} & a_{32} & a_{42} & \vdots & a_{n2} \\
c_{13} & a_{23} & a_{33} & a_{43} & \vdots & a_{n3} \\
c_{14} & a_{24} & a_{34} & a_{44} & \vdots & a_{n4} \\
c_{15} & \cdots & \cdots & \cdots & \cdots & \cdots \\
\cdots & \cdots & \cdots & \cdots & a_{ij} & \cdots \\
c_{1m} & a_{2m} & a_{3m} & a_{4m} & \vdots & a_{lm}
\end{array}\right)
$$

<Figure 3> IS context and pre-defined category in IS context memory

Total score and various weight are calculated as follows.

$$
\begin{aligned}
S_{k,j} &= \sum_{i=1}^{l} I_{i,j} \times a_{i,j} \\
TS_j &= \sum_{k=1}^{n} w_k \times S_{k,j}
\end{aligned}
\qquad (3)
$$

Where,
i stands for columns (category), $1 \le i \le l$
j stands for rows (context), $1 \le j \le m$
k means k^{th} document, $1 \le k \le n$
c_{1j} means context of IS ($i = 1$)
$S_{k,j}$ is score value of j^{th} IS context in k^{th} document
TS_j is total score value of j^{th} IS context
w_k is the weight of k^{th} document. The most recent document influences the determination of user context. w_k is a parameter and the following is an example. w_k is the weight for the first consideration that the existing document screen the user is browsing has an influence on determining user context rather than the preceding or later document screens. p is a user defined parameter

$$
\cdot \sum w_k = 1 \qquad w_k \left\{ \begin{array}{l} k = p \rightarrow x(x\rangle \frac{1}{k}) \\ k \neq p \rightarrow \frac{1-x}{k-1} \end{array} \right\} \qquad (4)
$$

$I_{i,j}$ is the weight of i^{th} category in j^{th} context. This weight reflects the second consideration that each context has a critical category which has an influence on user's context
$a_{i,j}$ is i^{th} category of i^{th} context. $a_{i,j}$ =1(lower level) or 1.1(upper level) or 1.2(top level) for selected, and $a_{i,j}$ =0 for non-selected. This weight reflects the third consideration that

upper level category is more weighted than lower level category to complement error of document categorization process.

4.3 Service inference algorithm

If user context is inferred, system can gather service information related to the inferred context from external environment through the internet and can constitute a deliver-enabled service list. Capturing user preference in advance, system can select user-adaptive services considering user preference and recommends them to the user. Therefore, to accomplish these, we need two assumptions.

- System perceives user preference in advance.
- System can extract service information technologically from other commercial web sites.

Based on these assumptions, the service inference algorithm is as follows

Step 1. Make deliver-enabled service list by using inferred context. Table 1 and 2 show the examples of user preference form and the deliver-enabled service list in the case of that the inferred context is a business trip and the system requires hotel and flight information.
Step 2. Calculate a similarity score between the user preference and the attributes in the deliver-enabled service list using formula (5).
Step 3. List user-adaptive services in descending order of the similarity score and deliver it to the user

The similarity score is calculated as follow.

$$S_{i,j,k} = \frac{\sum_{i=1}^{n}(w_i \times Sim(UF_i, SL_i))}{\sum_{i=1}^{n} w_i} + \frac{\sum_{j=1}^{m}(w_j \times UDF_j)}{\sum_{j=1}^{m} w_j} \tag{5}$$

Where,
S is a sum of the similarity score
C is a co-attribute set both user preference and deliver-enabled service list. C includes common attributes of the user preference and the deliver-enabled service list
B is a user selected attribute set in the deliver-enabled service list. B excludes attributes in C
i is a i^{th} attribute of the co-attribute set C. $1 \leq i \leq n$
j is a j^{th} attribute of the user selected attribute set B. $1 \leq j \leq m$
k is a k^{th} service in the deliver-enabled service list. $1 \leq k \leq 1$
w_i is a i^{th} attribute weight in set C . Critical attributes in the user preference are more weighed than other attributes.
w_j is a j^{th} attribute weight in set B. Important attributes in the deliver-enabled service list, which are not included in the user preference, are added to the similarity calculation process.
UF_i is a i^{th} attribute of the user preference
SL_i is a i^{th} attribute of the deliver-enable service list
UDF_j is a user-defined function for adding important attributes in deliver-enabled service to the similarity calculation process.
$Sim(UF_i\,SL_i)$ is a similarity function between the UF_i and the SL_i

<Table 1> Example of user preference form in the case that inferred context is a business trip and the system requires hotel and flight information

Hotel preference		Flight preference	
hotel grade	integer (1~5)	air lines	character
room type	1 : non-smoking, 0 : smoking	seat type	1 : window seat, 0 : aisle seat
room preference	L : low floor, M : middle floor, H : high floor	class type	1 : economy class, 0 : business class

<Table 2> Example of deliver-enabled service list in the case that inferred context is a business trip and the system requires hotel and flight information.

Flight service list

Company name	Seat type	Class type	price	Flight no.	Seat no.	Departure data
character	1 or 0	1 or 0	integer	integer	integer	Integer

Hotel service list

Hotel name	Hotel grade	price	Room type	Room preference	Location	Room no.
character	Integer(1~5)	integer	1 or 0	L or M or H	character	integer

Figure 4 shows examples of the Sim(UFi SLi) and the UDFj using the data from table 1 and 2

Flight information

$$Sim(UF_1, SL_1) = 1.5 - \left| companyname_{UF} - companyname_{SL} \right|$$

$$Sim(UF_2, SL_2) = 1.5 - \left| seattype_{UF} - seattype_{SL} \right|$$

$$Sim(UF_3, SL_3) = 1.5 - \left| classtype_{UF} - classtype_{SL} \right|$$

C = { company name, seat type, class type }

B = { $price$ } c

$$UDF_1 = \frac{price_{max} - price_{SL}}{price_{max} - price_{min}} + 0.5$$

Hotel information

C = { *hotel grade, hotel type, room preference* }

B = { *price* }

.

<Figure 4> examples of the $Sim(UF_i\, SL_i)$ and the UDF_j

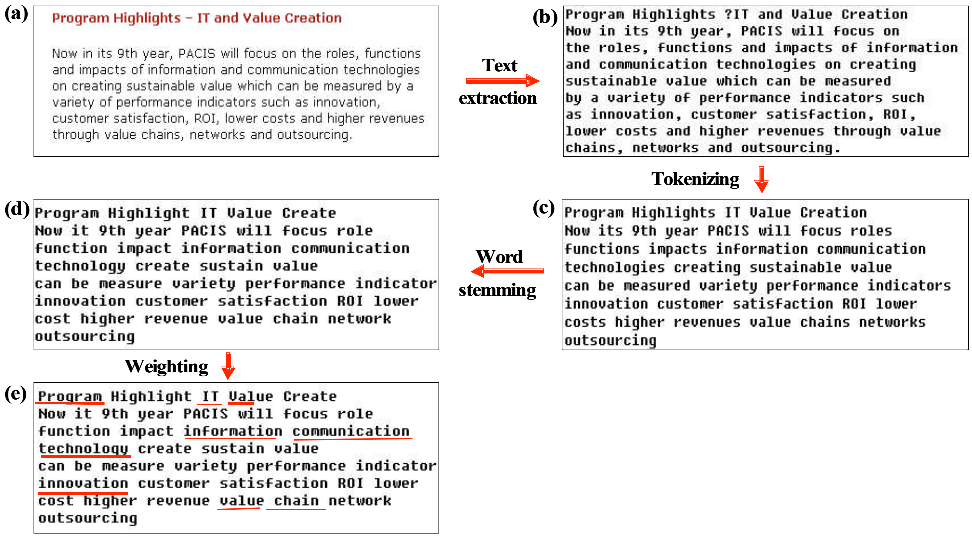
5. CASE STUDY FOR CONTEXT AND SERVICE INFERENCE

5.1 Simple case

As an example, we describe a situation when a user intends to request through an IS a business trip to attend a conference on management. The user is a male and he is a professor at a university. When traveling by airplane, he prefers an economy class and a window seat. Assume that he prefers Asiana Airlines to other Korean air carriers.

To request the business trip, he needs information about transportation and accommodation. The user browses a webpage in order to find the required information while logged into the IS. First, the user browses the conference's webpage to obtain where and when it will be held. Second, the user browses the Hilton hotel webpage to check room availability and the cost. Third, the user opens Korea Bank homepage to confirm his account balance.

The system recognizes user context (business trip request) by using the context inference algorithm, extracts the related information considering user preference, and recommends user-adaptive services related to the transportation and the accommodation by using the service inference algorithm.

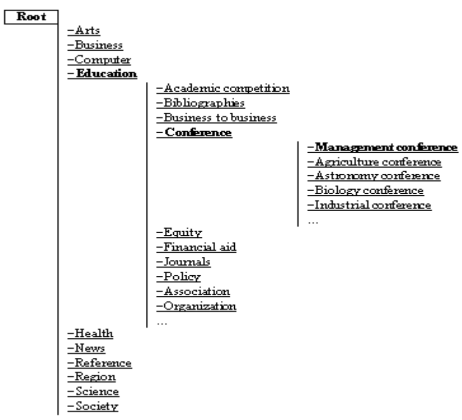


<Figure 6> An example of the preprocessing procedure, (a) Raw web page, (b) Texts extracted from the raw web page, (c) Text should be broken into discrete units, (d) Tokenized words should be standardized, (e) To solve the dimensionality problems, underlined words are selected and weighted

5.2 Representative extraction

The first step for inferring user context is the text categorization of each webpage that the user is browsing. As we mentioned before, for precise categorization, constructing the corpus is very important. To construct precise corpus, categories should be predetermined adequately. Therefore, we should analyze the menu structure and the contents of the IS. The results of the analysis must be reflected in the corpus construction. Figure 5 shows an example of category hierarchy in the corpus after analyzing the IS menu structure and contents.

The corpus and a new sample should be preprocessed by such procedures as tokenizing, word stemming, feature selection and weighting. Figure 6 shows an example of the preprocessing procedure. After the preprocessing, the classifier categorizes the new sample. The *k*-nn approach is used as a classifier. From the categorization process, we can obtain the results, which are 1st window : management conference, 2nd window : hotel and 3rd window : bank.



<Figure 5> Example of category hierarchy in corpus

5.3 Context inference

Category data of each webpage are used for recognizing user context by the context inference algorithm. Table 3 shows the IS contents and the pre-defined categories stored in the IS context memory. IS context is obtained by contents and the menu structure analysis of the IS which are executed in advance, and pre-defined categories are included in the corpus constructed in advance. Numbers in brackets mean the weight of a critical category that influences the context and these weights are user-defined parameters.

<Table 3> Table form of the IS context and the pre-defined category in IS context memory. Numbers in brackets mean the weight of a critical category.

Context	**Pre-defined category**
Business tour request	
Conference attendance	education (0.1), conference (0.3), management conference(0.1) business (0.1), accommodation (0.1), hotel (0.2), …
Business meeting	business (0.1), travel (0.1), airline (0.3), travel agency (0.1), accommodation (0.1), hotel (0.3), …
Dispatch work	business (0.1), financial (0.1), bank (0.2), …
…	…
Purchasing request	
Office appliance	office supplier (0.3), Shopping (0.1), …
...	…

The determined categories are management conference, hotel, and bank. Therefore, the full input category hierarchies are education/conference/management conference, business/accommodation/hotel, and business/financial/bank, as in figure 5. Because the user browses three windows, k value is equal to 3. The weight of k^{th} document, w_k, is determined by formula (4), where p is a user-defined parameter. Table 4 shows the values of w_k when k=2, p=0.6; k=3, p=0.5; k=4, p=0.4; k=5, p=0.4. In the example drawn earlier, k is equal to 3. Therefore, we uses $w_1=0.25$, $w_2=0.25$, $w_3= 0.5$.

< Table 4 > Values of w_k , where k=2, p=0.6; k=3, p=0.5; k=4, p=0.4; k=5, p=0.4

$k=2$	$w_1= 0.4$, $w_2=0.6$
$k=3$	$w_1=0.25$, $w_2=0.25$, $w_3= 0.5$
$k=4$	$w_1=0.2$, $w_2=0.2$, $w_3=0.2$, $w_4=0.4$
$k=5$	$w_1=0.15$, $w_2= 0.15$, $w_3=0.15$, $w_4=0.15$, $w_5=0.4$

Category hierarchy is calculated by formula (3). The calculation procedure and the results are showed in table 5

Following Table 5, the total scores are as follows: TS_1 =0.503, TS_2 =0.193, and TS_3 =0. 215. TS_1 is the biggest than other values, therefore, context of TS_1 , which is the business trip request to attend conference, is selected.

<Table 5> Calculation of total score

	Category hierarchy	a_{ij}	I_{ij}	*Score*
$k=1$	education/conference/ management conference	$a_{21}=1.2$, $a_{31}=1.1$, $a_{41}=1$	$I_{21}=0.1$, $I_{31}=0.3$, $I_{41}=0.1$	$S_{11}=0.55$
$k=2$	business/accommodation/hotel	$a_{51}=1.2$, $a_{61}=1.1$, $a_{71}=1$	$I_{51}=0.1$, $I_{61}=0.1$, $I_{71}=0.2$	$S_{21}=0.43$
		$a_{22}=1.2$, $a_{62}=1.1$, $a_{72}=1$	$I_{22}=0.1$, $I_{62}=0.1$, $I_{72}=0.3$	$S_{22}=0.53$
$k=3$	Business/financial/bank	$a_{51}=1.2$	$I_{51}=0.1$	$S_{31}=0.12$

		a_{22}=1.2	I_{22}=0.1	S_{32}=0.12
		a_{23}=1.2, a_{33}=1.1, a_{43}=1	I_{23}=0.1, I_{33}=0.1, I_{43}=0.2	S_{33}=0.43

5.4 Service inference

The system recognizes the user's cognitive context as a 'business trip request' by using the context inference algorithm. To a request business trip, the user needs information about transportation and accommodation. Therefore, the system extracts service information related to the transportation and accommodation from external commercial websites through the internet considering the user's destination and departure date. Table 6 shows the deliver-enabled service list extracted by the system from the websites.

<Table 6> Deliver-enabled service list

Company name	Seat type	Class type	price	Flight no.	Seat no.	Departure date
Asiana	1	0	300	A707	W12	2005.07.05. 13:00
Jal	1	1	350	J123	W33	2005.07.05, 13:15
Korean air	0	1	360	K677	A23	2005.07.05, 12:00
Asiana	0	0	280	A977	A77	2005.07.05, 14:00
Asiana	1	1	360	A677	W10	2005.07.05. 13:20
Air china	1	0	315	C344	W11	2005.07.05. 12:45
Air china	0	1	330	C756	A45	2005.07.05. 13:40
Asiana air	0	0	250	A209	A54	2005.07.05. 13:25
Korean	1	0	290	K667	W27	2005.07.05. 14:00
Asiana	1	1	305	A308	W30	2005.07.05. 13:10

Out of all the deliver-enabled services, the system selects user-adaptive services. To accomplish this, the system calculates the similarity between the user preference and the deliver-enabled service. Table 7 shows user preference about an air flight. In the table, '1' indicates 'seat type' as being 'window seat', and 'class type' as being 'economy class'.

<Table 7> User's air flight preference

Air Lines	seat type	class type
Asiana (Rep. of Korea)	1	1

The first step for calculating the similarity is to define the attributes set C and B. Because user's flight preference is composed of 3 attributes, which are Air Company, seat type, class type, set C includes the same attributes. In other words, c = {Air Company, seat type, class type}. If the price is important for the user, but it is not included in the attributes in set C, then, it should be included in set B. That is, B = {price}. Then, concerning the attributes in set C, system calculates $Sim(UF_i, SL_i)$ by using the formula shown in figure 4. Concerning the attributes in set B, the system calculates UDF_j in the same manner. After calculating $Sim(UF_i, SL_i)$ and UDF_j, the system computes the total similarity by using formula (5) Table 8 shows the calculation of similarity

<Table 8> The calculation of similarity

Flight no.	$Sim(UF_1\ SL_1)$	$Sim(UF_2\ SL_2)$	$Sim(UF_3\ SL_3)$	UDF_1	Total similarity
A707	1.5	1.5	0.5	1.04545	2.628571
J123	0.5	1.5	1.5	0.59091	2.742857
K677	0.5	0.5	1.5	0.5	1.371429
A977	1.5	0.5	0.5	1.22727	1.257143
A677	1.5	1.5	1.5	0.5	3
C344	0.5	1.5	0.5	0.90909	2.371429

C756	0.5	0.5	1.5	0.77273	1.371429
A209	1.5	0.5	0.5	1.5	1.257143
K667	0.5	1.5	0.5	1.13636	2.371429
A308	1.5	1.5	1.5	1	3

The higher the value of the total similarity is, the closer the service is to the user preference. Therefore, the system lists the user-adaptive services in the descending order of total similarity and delivers them to the user. Table 9 shows the user-adaptive service list in the case when the user sets the value of the total similarity more than 2.

<Table 9> User-adaptive service list

Flight no.	Company name	Seat type	Class type	Seat no.	Departure data	price	Total similarity
A677	Asiana	1	1	W10	2005.07.05. 13:20	360	3
A308	Asiana	1	1	W30	2005.07.05. 13:10	305	3
J123	Jal	1	1	W33	2005.07.05, 13:15	350	2.742857
A707	Asiana	1	0	W12	2005.07.05. 13:00	300	2.628571
C344	Air china	1	0	W11	2005.07.05. 12:45	315	2.371429
K667	Korean	1	0	W27	2005.07.05. 14:00	290	2.371429

6. CONCLUSION AND FUTURE WORK

This paper proposes context and service inference algorithms for a web-based IS domain. The context inference algorithm aim is to recognize the intentions of a user working within an IS. To apply this algorithm, we used a text categorization technique to classify various text-based sources a user browses such as web pages, PDF-documents, MS-word documents, etc. The service inference algorithm recommends user-adaptive service and is based on the similarity measurement between the user preference and the deliver-enabled services. The study also demonstrated the operations of the two algorithms through a simple case study.

Existing studies on the context-awareness obtain context information physically by using stereo type cameras, RFID, smart devices, etc. However, there have been no trials to infer a user’s context in a static condition. In other words, research on inferring cognitive context has not yet been performed. In this fashion, the attempt to infer user’s cognitive context in the information system domain is unique. Therefore, the context and the service algorithm this paper proposes this paper proposed will help users work with ISs conveniently and enable existing ISs deliver ubiquitous service. Moreover, this paper shows the direction that existing ISs could head, namely, towards a ubiquitous-enabled IS. However, we need to expand the algorithms into more practical situations. The algorithms that we demonstrated operated on several assumptions. In our future endeavors on this project we will improve the algorithms to remove these assumptions and make them robust enough for real world situations. Additionally, we will on develop and evaluate a working prototype in order to validate our algorithms.

References

Abowd G. D., Dey A. K., Brown P. J., Davies N., Smith M. and Steggles P., “Towards a Better Understanding of Context and Context-Awareness”, Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, Karlsruhe, 1999, Germany

Asirvatham A. P. and Ravi K. K., "Web Page Classification Based on Document Structure", IEEE National Convention, 2001
Brown P. J. and Jones G. J. F., "Context-aware Retrieval: Exploring a New Environment for Information Retrieval and Information Filtering", Personal and Ubiquitous Computing 5, 2001, P 253-263
Brumitt B. and Meyers B., "EasyLiving: Technologies for intelligent environments", In Proceedings of 2nd International Symposium on Handheld and Ubiquitous Computing (HUC), 2000, Bristol, UK
Gwizdka J., "What's in the context?", Computer Human Interaction(CHI), Hague, Netherlands, 2000
Khedr M. and Karmouch A., "ACAI: agent-based context-aware infrastructure for spontaneous applications", Journal of Network and Computer Applications 28, 2005, P 19-44
Ko Y. J., Park J. W. and Seo J. Y., "Improving text categorization using the importance of sentences", Information Processing and Management 40, 2004, P 65-79
Kwon O. B. and Sadeh N., "Applying case-based reasoning and multi-agent intelligent system to context-aware comparative shopping", Decision Support Systems 37, 2004, P 199-213
Kwon O. B., Yoo K. D. and Suh E. H., "UbiDSS: a proactive intelligent decision support system as an expert system deploying ubiquitous computing technologies", Expert Systems with Applications 28, 2005, P 149-161
Kwon O. W. and Lee J. H., "Text categorization based on k-nearest neighbor approach for Web site classification", Information Processing and Management 39, 2003, P 25-44
Pierre J. M., "On the Automated Classification of Web Sites", LinkÖping Electronic Articles in Computer and Information Science Vol. 6, 2001
Prekop P. and Burnett M., "Activities, context and ubiquitous computing", Computer Communications 26, 2003, P 1168-1176
Ranganathan A. and Campbell R. H., "A Middleware for Context-Aware Agents in Ubiquitous Computing Environments", In ACM/IFIP/USENIX International Middleware Conference, 2003, Rio de Janeiro, Brazil
Schilit B. N., Adams N. and Want R., "Context-Aware Computing Applications", In Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, 1994, Santa Cruz, CA., USA
Schilit B. N., Adams N., Gold R., Tso M. and Want R., "The PARCTAB Mobile Computing System", In Proceedings of the Fourth Workshop on Workstation Operating Systems (WWOS-IV), 1993, Napa, CA, USA
Schmidt A., Beigl M. and Gellersen H. W., "There is more to context than location", Computers and Graphics 23, 1999, P 893-901
Sebastiani F., "Machine Learning in Automated Text Categorization", ACM Computing Surveys 34, 2002, P 1-47
West D., Apted T. and Quigley A., "A context inference and multimodal approach to mobile information access", Artificial Intelligence in Mobile Systems (AIMS), 2004, Nottingham, England
Yang Y. and Liu X., "A re-examination of text categorization methods", International Conference on Research and Development in Information Retrieval (SIGIR), 1999, Berkley, CA, USA
Yang Y., Slattery S. and Chain R., "A study of approaches to hypertext categorization", Journal of Intelligent Information Systems 18, 2002