

A context-aware search system for Internet of Things based on hierarchical context model

Yuanyi Chen · Jingyu Zhou · Minyi Guo

Published online: 26 March 2015
© Springer Science+Business Media New York 2015

Abstract In recent years, numerous sensing devices and wireless networks are immersed into our living environments, creating the Internet of Things (IoT) integrating the cyber and physical objects. Searching for objects in IoT is a challenging problem because the context relationships among IoT objects are various and complex. The traditional web search approaches cannot work well in the IoT search domain because they miss the critical characteristics of the context relationships. In addition, a user's dynamic and changing context affects the user's information needs, and an IoT search system should exploit context relationship in IoT for retrieving relevant information suitable for the user's current context. In this paper, we present a context-aware search system for IoT, which aims to search objects and related information with more suitable results. We construct a hierarchical context model based on ontology to represent the IoT objects and their contextual relationships. Then, searches are executed with consideration of users' context that is recognized by a context-aware hidden Markov model. Experimental results confirmed that users could obtain more suitable and reasonable search results than with a typical web or map search system.

Keywords Internet of Things · Context model · Searching · Context recognition

Y. Chen · J. Zhou (✉) · M. Guo
Embedded & Pervasive Computing Lab, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
e-mail: zhou-jy@cs.sjtu.edu.cn

Y. Chen
e-mail: scu_cyy@163.com

M. Guo
e-mail: guo-my@cs.sjtu.edu.cn

1 Introduction

The Internet of Things (IoT) will be formed with the rapid development and applications of low-cost embedded sensing devices that people use in daily life (such as mobile phones, cars, wallets and key-chains, etc) will have the ability of sensing and communication in the near future. Zhang et al. [1] predicted that IoT eventually will link the majority of objects into the virtual space and allow objects to interact in the same place. Currently, search is moving from the traditional concept of finding documents, pictures or music to all aspects of life and work. Searching information in IoT is becoming an indispensable application.

Unfortunately, the algorithms designed for Internet searching are not suitable for the IoT search because of the following two reasons. Firstly, the relationships among the IoT objects are diverse and complex (such as geographical link, behavior link and social network link, etc.), thus the relationship cannot be represented by simple universal resource locator (URL) approach. Secondly, the information of physical objects is different from the web pages in terms of the meta-data and information dynamics. New searching algorithms are needed for gathering, extracting and organizing the information in IoT.

To extend the capability of searching information in IoT, researches have worked on alternatives to traditional Internet search methods for the IoT environment. Early work explored locating daily objects using a variety of sensing technologies [2–7] (e.g., RFID tracking, radio frequency, Bluetooth or ultrasound) to collect the location information of objects into a server and retrieve location information from the server. Some other studies have extended searching capability from location to other information in a small-scale environment [8–11]. However, all these previous systems do not consider

the context information of the IoT objects, which plays a significant role in IoT search [8].

Context is an inaccurate and broad concept. Dey and co-workers [12] define context as “Any information that can be used to characterize the situation of an entity. An entity is a user, a place, or a physical or computational object that is considered relevant to the interaction between a user and an application, including the user and application themselves.” Context information such as the current time or user’s location among the IoT objects has been utilized in some recent studies [13–16]. However, the context information in these studies is limited and does not capture various and complex relationship among IoT objects. Therefore, this kind of system has limited utility in practice.

Moreover, all of the above studies have failed to consider the user’s context. Recognition of users’ context can assist search system for context reasoning and improving the effectiveness and relevance of search results. For instance, mobile users’ searches frequently change in their contexts. The users’ context includes actions (such as walking, running or sitting) and activities (such as eating, attending class at morning or going to airport). The user’s context can affect their information needs. When a user needs to reach airport as soon as possible due to the tight boarding schedule and he searches the possible transport routes to airport, time is more important than cost and other factors in the context. In this scenario, the search results should return the fastest route.

In this paper, we present the design and implementation of a context-aware search system for IoT based on hierarchical context model. Firstly, a hierarchical context model based on ontology was built to describe the IoT objects and their contextual relationships. Specifically, the model uses Web Ontology Language (OWL) to represent various objects and object contextual relationship. Secondly, effective context reasoning by considering the users’ context is used to aid effective information searches. The user’s context is derived from activity and other supplementary information (such as location and time, etc.), where activities are recognized by a context-aware hidden Markov model (HMM). Finally, the system will retrieve return information in context database based on OWL description logic query.

Our experiments confirmed that participants can get more suitable search results than commercial search systems and can answer some new types of searches that current commercial search system cannot do. Experimental results also demonstrate that the activity recognition method in the paper has better performance than pure statistical methods.

The remainder of this paper is organized as follows: Sect. 2 describes related works. Section 3 presents the design and implementation of our system. Section 4 discusses our experimental results. Finally, Sect. 5 draws a conclusion for this paper.

2 Related work

In this section, we survey some previous studies that address searching information in IoT and discuss how these studies differ from our work. In general, these existing search systems for the IoT objects can be divided into two categories:

(1) *Search system for IOT without consideration of context.* This kind of systems only provide some simple location search by embedded devices that contain pre-defined description of objects, without modeling different IoT objects and considering their contextual relationships. For example, SensorWeb [3] aimed to resource sharing and integration of ubiquitous devices and data; Max [6] provided a three-tier centralized hierarchy search engine for object searching. The hierarchical architecture consists of tags, sub-stations and base-stations. Tags can be marked as either public or private with private tags searchable only by the owner. Substations can sense the objects nearby by RFID embedded, they are responsible for building an inverted index of their nearby objects; TinyDB [11] used wireless sensor network for querying the physical object values and address the issue of how to get the pre-defined, homogeneous information from a small scale static environment. The base station stores the inverted index of substations. Snoogle [13] supported multiple-keyword search and top-k query by adding information retrieval techniques to index information and process user queries, and bloom filters to reduce communication overhead. Microsearch [14] extended Snoogle for more details about the design and implementation of the top-k search and presents a memory efficient algorithm and a theoretical model of the search.

(2) *Search system for IOT with consideration of context.* This kind of systems only consider limited contexts such as location or time by embedding objects with ubiquitous sensors, without considering a user’s context. Context-aware is a fundamental feature of IoT, context relationship should be considered for these search systems whose target are more than just location and find physical objects in IoT. Heil et al. [17] proposed a location-based approach for federating services and devices. Comparing to existing approaches of location-based and geo-centric services, this system connects the functionality of devices corresponding to their physical distance without considering their ownerships. Ostermaier et al. [18] proposed a real-time search engine for the web of things, which extended the search function to support rapidly changing state information generated by sensors. Perera et al. [8] proposed a context-aware sensor search model for the IoT middleware to address the research challenges of selecting sensors when large numbers of sensors with overlapping and sometimes redundant functionality are avail-

able. This system searches and selects sensors based on user priorities by using both semantic querying and quantitative reasoning techniques. Shen et al. [19] proposed a social-aware distributed cyber-physical human-centric search engine for IoT to address the difficulty of widely deploying RFID devices and the centralized search. This system also locates objects held by users based on the routine user movement pattern and using a social-aware Bayesian network to predict the users' locations when exceptional events.

Recently, building context model for objects of IoT is under active research. Ejigu et al. [15] proposed an ontology-based generic context management model for modeling and reasoning contexts. Hung et al. [16] proposed a context-aware middleware for pervasive elderly home-care with a P2P-based context query processing and reasoning. Hervás et al. [20] proposed a context model based on ontological languages. All previous models are based on ontology with the advantages of interoperability and heterogeneity, but ontology-based models are time-consuming to reasoning [21]. For search based on user's context, Maekawa et al. [22] proposed a context-aware web search method which can automatically retrieves a webpage related to a user's daily activity, which is detected with object-attached sensors. However, these systems consider limited contexts by embedding objects with ubiquitous sensors, without considering a user's context when he/she search. In contrast, we focus on searching in IoT with considering users' search context.

3 Design

The section first gives an overview of our proposed system, and then presents the details of our hierarchical context model.

Figure 1 shows the high-level overview of our system. Context database represents instances of context that may exist in the form of profiled data stored on a disk file or in the form of context instances obtained from the sensors. Semantic rules database represents instances of reasoning rules that are used by context reasoning to reason out and derive decisions. The rules include implicit rules derived by ontology and explicit rules by humans.

The system consists of two basic components: semantic interpreter and context reasoning. Semantic interpreter processes search input and the user's current context obtained from the user's context recognition component to provide search key words and information for context reasoning. Specifically, a user's context is recognized based on activity recognition and other useful information (e.g., the current time, the user's location, the user's search history, the user's schedule or profile). The context reasoning compo-

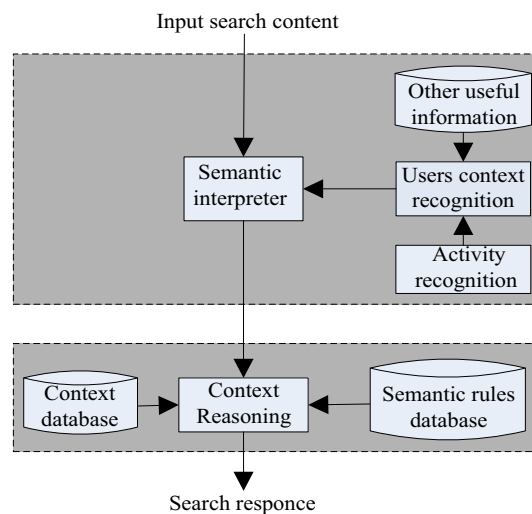


Fig. 1 High-level overview of the system

nent implements the proposed hierarchical context model to make context reasoning based on context database and reasoning rules obtained from semantic rules database. Finally, the system will return search response information to the user from context database based on context reasoning.

3.1 User's context recognition based on a context-aware hidden Markov model

In our system, we recognize users' context in two processes: (1) to recognize users' activity based on a context-aware hidden Markov model; (2) to combine other context information to infer users' context.

3.1.1 HMM for activity recognition

Roughly speaking, activity recognition methods can be classified into two categories: (1) recognition methods based on supervised learning algorithms, such as [23–25]. Those methods need to be trained with a sufficient set of labeled samples in order to perform well, while labeled samples are very hard to acquire due to many factors (such as cost, time and user privacy, etc.). Therefore, methods based on supervised learning algorithms are prone to serious scalability issues when more activities are considered. Even if enough train samples exist, it is worth to note that most models adopted by supervised learning algorithms implicitly assume independence between each pair of instances to be classified. However, when considering activity instances, the assumption does not hold. In fact, persons do not continuously switch among different activities; instead, they tend to perform the same activity for a certain lapse of time before changing. (2) Recognition methods based on self-unsupervised learning algorithms, such as [26–29], large train data is unnecessary for this kind of methods.

Human activity recognition has the following two characteristics: (1) the collected sensor data for human activity is time series data (such as accelerometer and gyroscope), which is using for extracting feature for activity recognition; (2) the probability from one activity transfer to another is relatively stable in given context. For instance, the students of upper floors who go out of the classroom have a high probability to walk down stairs while not walk up stairs after class. In view of the above two characteristics, our system chooses HMM to recognize human activity. Let $Q = \{q_1, q_2, \dots, q_N\}$ is the set of all possible states, $V = \{v_1, v_2, \dots, v_M\}$ is the set of all possible observations, then a HMM can be definition as a 3-triple, as Eq. (1) shows:

$$\lambda = (A, B, \pi), \tag{1}$$

where $A = [a_{ij}]_{N \times N}$, represent the state transition probabilities, and a_{ij} is the probability from state i transfer to state j ; $B = \{b_i(k)\}$ represent for the observation probability distribution, and $b_j(k) = P(o_t = v_k | i_t = q_j)$ is the probability of generation observation v_k in the state of q_j and at time t ; $\pi = \{\pi_i\}$ represent the initial state distribution.

In this article, we proposed an activity recognition model based on HMM, the structure shows in Fig. 2.

In our approach, the HMM for activity recognition is discrete rather than continuous for reducing additional computation, using K-means for quantization. Figure 2 shows the HMM structure for activity recognition. Firstly, build HMM based on train data with the form of $\lambda_i = (A_i, B_i, \pi_i)$. Then for a given observation activity sequence: $O = (O_1, O_2, \dots, O_T)$, using Baum–Welch algorithm [30] to find λ_i that make $P(O|\lambda_i)$ maximum as the inferred activity.

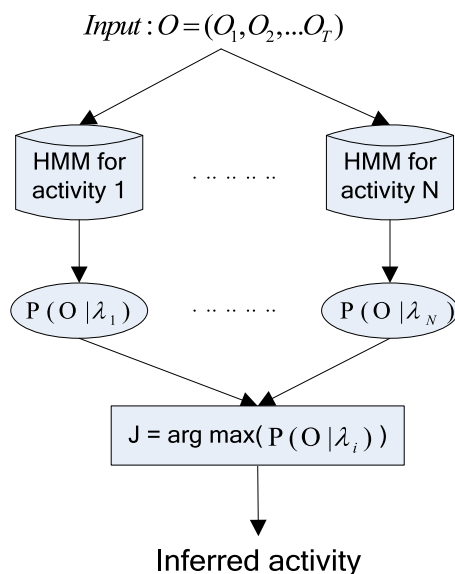


Fig. 2 Activity recognition based on HMM

Although recognition methods based on HMM consider the dependency relationship between each pair of instances, HMM assume the probability from one activity transfer to another is fixed, when considering activity affects by lots of contextual information the assumption does not hold. For instance, the transfer probability between two activities is obvious variable in different time and location. One of the main limitations of the HMM method is that it did not consider contextual information (e.g., location, time and surrounding objects) that could be usefully exploited to derive the user’s activity [31]. As a consequence, our system was aimed at devising activity recognition by taking into account the user’s context.

3.1.2 Context-aware HMM combination with users’ context

Assume there have N kinds of activities need to be recognized in total, for an input activity instance i , the HMM model will predict $P(s_j|i)$, which means the probability that the input activity instance i recognized as activity s_j . Define $\vec{v}_i = P(s_1|i)P(s_2|i) \dots P(s_N|i)$, then the HMM model will achieve a N-length probability vector \vec{v}_i , for a single activity instance i , in which the j th element $\vec{v}_i^{(j)}$ corresponds to activity s_j and its value corresponds to the probability of the recognition model regarding the association of instance i to activity s_j , obviously $0 \leq \vec{v}_i^{(j)} \leq 1$ and $\sum_{j=1}^N \vec{v}_i^{(j)} = 1$.

For instance, suppose that the considered activities are those shown in Table 1, there have four activities in total (walking up stairs, walking down stairs, sitting and standing). Assume a student is walk up stairs in teaching building for attending class at 7:30. In this case, the maximum probability value (0.45) corresponds to walk down stairs, followed by walk up stairs (0.41), sitting (0.14) and standing (0.0). Hence, according to the criterion that select the maximum probability, the HMM would erroneously infers that the student’s activity is walk down stairs. The reason is recognition model based on HMM alone will draw erroneous conclusion with little difference among the inferred probabilities of some activities.

In contrast, people perform well when encountering this situation by combination other useful context information [32,33]. Consider the instance mentioned previously, people can automatically infer that walk up stairs should be more reasonable activity based on the context that the student is

Table 1 HMM probability for one instance

Activity	Walking up stairs	Walking down stairs	Sitting	Standing
HMM probability	0.41	0.45	0.14	0.0

Table 2 Part of probability table for activities in specific location and time

Context	Activity			
	Walking up stairs	Walking down stairs	Sitting	Standing
Breakfast time (6:00–8:00)				
Teaching building	0.53	0.21	0.26	0.0
Laboratory	0.31	0.35	0.26	0.08
Canteen	0.23	0.13	0.43	0.21
Playground	0.15	0.13	0.55	0.17
Class time (8:00–12:00)				
Teaching building	0.16	0.11	0.73	0.01
Laboratory	0.04	0.13	0.78	0.05
Canteen	0.16	0.12	0.57	0.14
Playground	0.16	0.11	0.54	0.19

going to class in the teaching building according to time. Additionally, location and time are critical context information that can help to infer reasonable activity. Considering the example, we can construct a probability table that an activity may be performed in specific time and location according to the train data, as show in Table 2.

Considering the two kinds of probability: (1) from probability table of activities in specific location and time; (2) from the inferred probability vector \vec{v}_i of HMM, and both are assigned to the same probability weight 0.5. For instance, assume the inferred result of a user activity in teaching building at breakfast time base on HMM model is shown in Table 1, then according to Table 2, the final recognition results are as follows: the maximum value ($0.47 = 0.41 \times 0.5 + 0.53 \times 0.5$) corresponds to walk up stairs, followed by walk down stairs ($0.33 = 0.45 \times 0.5 + 0.21 \times 0.5$), sitting ($0.13 = 0.14 \times 0.5 + 0.26 \times 0.5$) and laying ($0.0 = 0.0 \times 0.5 + 0.0 \times 0.5$). Hence, we can infer the current activity is walking up stairs.

Figure 3 shows the algorithm for building probability tree for activities in specific time and location, which is executing by the offline. The algorithm takes four variables as input, $TDset$ represent for all the train instances set, S represent for all the activities label set, T represent for all the time periods set, L represent for all the locations set. The algorithm will calculate the probability for each activity in specific location and time and build a probability tree. For instance, the BP-tree algorithm will build a probability tree shown in Fig. 4.

In Fig. 4, a probability tree with three layers (1st layer is time, 2nd layers is location and 3rd layer the activity probability) was built by BP-tree algorithm. Note that time period partition and select locations are based on the specific situation, in our experiment the specific situation is school.

Combining HMM and BP-tree, our proposed context-aware recognition method for users' situation is shown in Fig. 5. Note that we recognize user's activity with a weighted sum approach based on online output prediction of HMM and

offline value of BP-tree. We choose different values of α to achieve the best performance in experiment.

3.2 Context model based on ontology for IoT

Our proposed system uses context model based on ontology to represent the objects and their context relationships in IoT.

3.2.1 A hierarchical context model based on ontology

Obviously, construct an exhaustive and detailed context model that represent all context information and relationship of objects in IoT is in-surmountable and unnecessary. Because some context (e.g. location, time, device, service, person, event, etc.) are fundamental and essential, we propose a hierarchical structure context model, where the upper context are these fundamental contexts, while the lower context is subclass of upper context sharing the same set of properties of upper context. Note that this hierarchical model can be easily adapted to other domains by inheriting the suitable upper context.

Figure 6 shows the partial inheritance structure of our context model. The first layer of our context model only has one root class named Things. The second layer of our context model is several subclasses that extend Things have been identified: Person, Activity, Device, Service, Location and Time, these abstract classes describing a physical or conceptual object. Each class is associated with its properties (represented in *DataProperty*) and relations with other classes (represented in *ObjectProperty*). The property *subClassOf* provides extension to new class. Figure 7 shows the partial *ObjectProperties* structure of our context model. The third layer of our context model is specific description of each class in the second layer, and the details will be given in the following.

Person our proposed context model is person-centered and hence the class "Person" is the central class in the model.

Fig. 3 Algorithm for building probability tree

BP-tree Algorithm

Input: $TDset, S, T, L$

Specially, $TDset$ is all the train instances set, S is all the activities label set, T is all the time periods set, L is all the locations set.

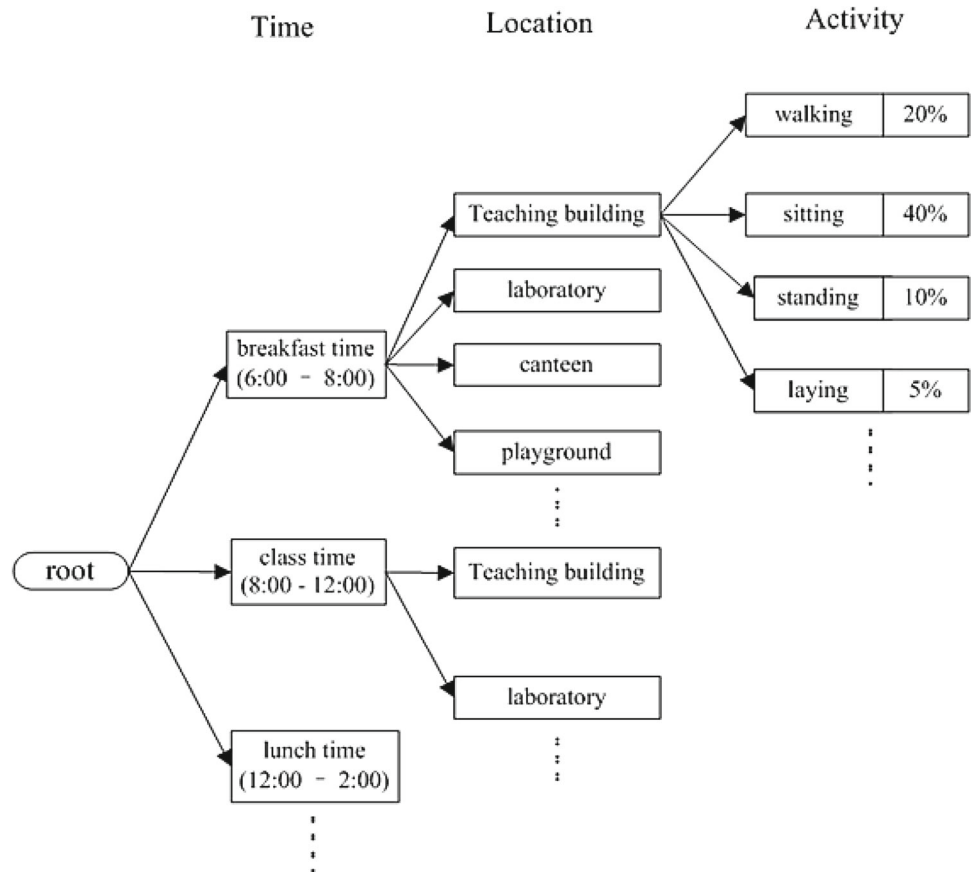
Output: The probability tree $BPTree$

```

1:  $BP-Tree(TDset, S, T, L)$ :
2:    $BPTree := buildTreeNoLeaf(T, L)$ ;
3:   For all instances  $a_i \in TDset$  do
4:      $p_i := Find(a_i \rightarrow time \in T)$ ;
5:      $q_i := Find(a_i \rightarrow location \in L)$ ;
6:      $calcCount(T_{p_i}, L_{q_i}, a_i)$ ;
7:   End
8:   For all elements  $p_i \in T$  and  $q_i \in L$  do
9:      $value_i := calcCount(T_{p_i}, L_{q_i}, s_i) \text{ DIV } calcCount(T_{p_i}, L_{q_i})$ ;
10:     $BPTree.addNode(s_i, value_i)$ ;
11:  End
12:  return  $BPTree$ :

```

Fig. 4 A probability tree builds by BP-tree algorithm



The “Person” class has several subclasses, such as “Student” and “Teacher”, etc. Moreover, the “Person” class offers var-

ious *DataProperties* and *ObjectProperties* (e.g. “ownedBy” is the *ObjectProperty* between person and device.) for incor-

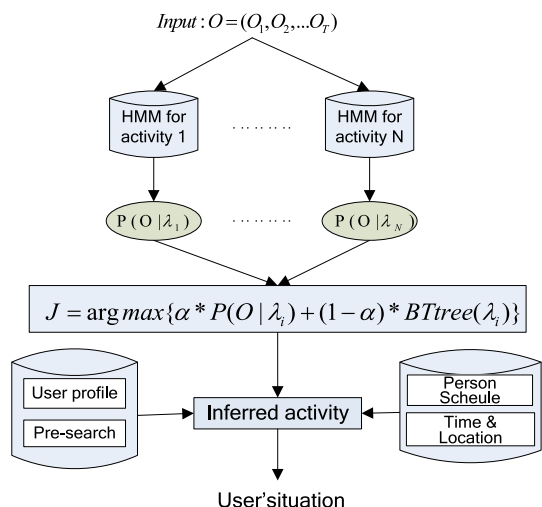


Fig. 5 User’s context recognition based on a context-aware HMM

porating the user’s related context. Table 3 shows the partial structure of person class.

Activity In order to include all information relevant to the person’s activity, The “Activity” class offers *DataProperties* such as “Activity Profile” to describe the related information of activity itself and *ObjectProperties* such as “beDefined” to describe the goal of activity. Table 4 shows the partial structure of activity class. Moreover, the “Activity” class also has several subclasses, such as “Eating”, “Meeting” and “Shopping”, etc.

Device the device (such as sensors or software device) is essential to collect context information and recognize the person’s context as we pointed out previously. The “Device” class has several subclasses, such as “Soft_Device” and “Hard_Device”, etc. The “device” class also offers various *DataProperties* and *ObjectProperties*, such as the DataProp-

erty “dProfile” to hold various data type properties representing configuration features. Table 5 shows the partial structure of device class.

Service the “Service” class captures information relevant to the activity and device (e.g. using the *DataProperties* “beDefined” to describe the schedule of a service and using the *ObjectProperties* “beProvidedBy” to illustrate the service can be provided by specific device.). The “Service” class has some subclasses, such as “Self-study” and “Dinner”, etc. Table 6 shows the partial structure of service class.

Location the location class represents the abstraction of a physical place with providing a set of *DataProperties* that associate a physical location with its symbolic or geographic location. Our proposed model regards location as an independent based on the point that location is fundamental context information for other classes. The “Location” class has some subclasses, such as “Classroom”, “Library” and “Canteen”, etc. Table 7 shows the partial structure of location class.

Time time is crucial context information just as location. The time context comprises all information related to the current time and serves as a timestamp for all context information that may change over time. Table 8 shows the partial structure of time class.

In general, the proposed context model is built upon the fact that a class, which corresponds to a physical object or conceptual object in IoT. The proposed context model is not tied to a specific domain. Instead, it attempts to describe generic concepts and their relationships.

3.3 Context reasoning

Auto-reason is the most important feature of ontology and context that can be implemented with logical reasoning

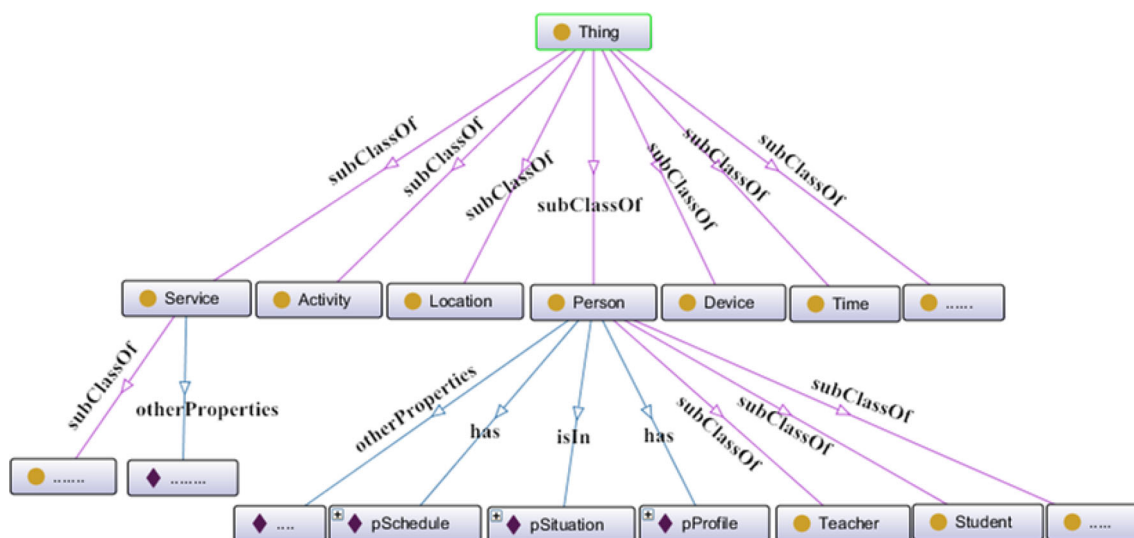


Fig. 6 The hierarchical structure of context model

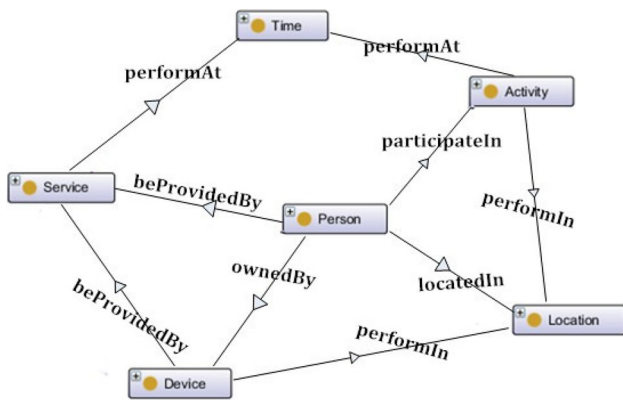


Fig. 7 The partial ObjectProperties structure of our context model

Table 3 The partial structure of person class

Properties name	pProfile	pSituation	pSchedule	Other properties name
Properties relationship	has	isIn	has	Other relationship

Table 4 The partial structure of activity class

Properties name	aProfile	aGoal	aPlan	Other properties name
Properties relationship	has	beDefined	has	Other relationship

Table 5 The partial structure of device class

Properties name	dProfile	dStatus	dSchedule	Other properties name
Properties relationship	has	has	beDefined	Other relationship

Table 6 The partial structure of service class

Properties name	sProfile	sSchedule	Other properties name
Properties relationship	has	beDefined	Other relationship

mechanisms by taking rules. Our system used Protege¹ to build context model based on ontology and FaCT ++² as the reason engine.

Rules are essential in the process of context reasoning. In our system, two forms of rules are used to context reason: implicit rules derived from the ontology and explicit rules provided by user in the specific application. Table 9 shows part reasoning rules of ontology.

Table 7 The partial structure of location class

Properties name	LProfile	Area	LSchedule	Other properties name
Properties relationship	has	has	beDefined	Other relationship

Table 8 The partial structure of time class

Properties name	currentTime	endTime	DuringTime	Other properties name
Properties relationship	has	has	has	Other relationship

4 Experiment evaluation

4.1 Situation recognition based on context-aware hidden Markov model

In order to evaluate our solution for activity recognition, we performed an experimental evaluation comparing our technique with three statistical methods (Bayesian network, KNN and SVM). Note that the context information time and location are used as features. For the context-aware HMM technique, time and location are not used as features but are used to build BP-tree. For our approach, the number of states for HMM is set to be four, and α is set to 0.3, unless specified otherwise.

4.1.1 Data collection

The smart-phone called HTC Sensation based on Android OS 4.0.3 was used as a platform for data collection. Participants were three PhD students between 25 and 30 years old. As shown in Table 10, each student performed six activities (walk, walk up stairs, walk down stairs, sit, stand and run) at four different locations (teaching building, laboratory, canteen and playground) in school. Using the embedded accelerometer and gyroscope of smart-phone, we captured 3-axial linear acceleration and 3-axial angular velocity. These sensor signals were pre-processed by applying noise filters.

The dataset is composed of 10,040 activity instances and the number of each category of activity in the same location is equal. The dataset is divided into two parts: training set (66%) and test set (34%), as shown in Table 11. For each activity instance, sensor data were merged to build a feature vector composed of 376 features, including means, variances, entropy, kurtosis, skew-ness and other statistical measures.

Table 9 Part reasoning rules

Category	Rule
Implicit (derived by Ontology)	Transitive Property: $(?A \text{ property } ? B) \cap (?B \text{ property } ? C) \Rightarrow (?A \text{ property } ? C)$ Symmetric Property: $(?A \text{ property } ? B) \Rightarrow (?B \text{ property } ?A)$ Inverse Property: $(?A \text{ propertyOne } ? B) \Rightarrow (?B \text{ propertyTwo } ? A)$
Explicit (pre-defined by user)	Meeting: $\left\{ \begin{array}{l} (?teacher \text{ schedule inf } o : \text{hasMeeting}) \\ (?teacher \text{ location in } : \text{meetingRoom}) \\ (?class \text{ time inf } o : \text{startTime} < \text{currentTime} \cap \text{endTime} > \text{currenttime}) \\ (?printer \text{ device inf } o : \text{statusOpen}) \end{array} \right. \Rightarrow (?teacher : \text{participant in? meeting})$ Silent Mode: $\left\{ \begin{array}{l} (?student \text{ schedule inf } o : \text{hasClass}) \\ (?student \text{ location in } : \text{classRoom}) \\ (?class \text{ time inf } o : \text{startTime} < \text{currentTime} \cap \text{endTime} > \text{currenttime}) \end{array} \right. \Rightarrow (?smartphone : \text{changeMode ?silent})$

Table 10 Experimental datasets for activity recognition

Activity location	Walk	Walk up stairs	Walk down stairs	Sit	Stand	Run
Teaching building	416	343	375	465	435	476
Laboratory	416	343	375	465	435	476
Canteen	416	343	375	465	435	476
Library	416	343	375	465	435	476

Table 11 Train and test dataset for activity recognition

Activity	Walk	Walk up stairs	Walk down stairs	Sit	Stand	Run
Total	1664	1372	1500	1860	1740	1904
Train_Set	1072	908	980	1210	1167	1289
Test_Set	592	464	520	650	573	615

4.1.2 Evaluation of activity recognition based on context-aware HMM

In this experiment, we performed tenfold cross validation on train dataset to construct activity recognition model and used test dataset to evaluate performance. Ideally, an out-of-the-box activity recognition system should be able to recognize one person’s activities without the need of being trained on that person. Hence, in order to avoid the use of activity data of the same user for both training and testing we ensured that activity instances regarding a given student did not appear in more than one sub sample.

Since there is no simple theoretically correct way of choosing the number of states for HMM, we determine the best

number of states with experiments and used the Forward algorithms to compute the various likelihood functions. The best classification result of 87.8 % is obtained when the number of states is four, as shown in Fig. 8. This reveals that in HMM classifier, the higher number of states does not necessarily imply better performance.

Figure 9 shows the result of HMM comparison with three statistical methods (Bayesian network, KNN and SVM). Recognition for ‘run’ activity show good performance except for ‘sit’, ‘stand’, ‘walk’, ‘walk up stairs’, ‘walk down stairs’ action and we will discuss the reason in the later section. The average precision for the six kinds of activities is shown in Table 12, the HMM method shows better average performance than the other three methods.

Note that we use try-and-error method to determine the value of α , where $\alpha = [0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]$, the recognition performance with different values of α can be seen in Fig. 10. Figure 11 shows the result of HMM comparison with context-aware HMM (according to Fig. 10, the precision is highest when $\alpha = 0.7$). The context-aware HMM method shows better performance than purely HMM method with precision 92.9 and 87.8 %, respectively. The confusion matrix of HMM method and context-aware HMM method are shown in Tables 13 and 14, respectively. It can be found from Table 13 that classification between ‘stand’ and ‘sit’ are confusing, so does among ‘walk’, ‘walk up stairs’ and ‘walk down stairs’. It meant that classification these activities are very difficult by using built-in sensors on a smart-phone. It can seen from Table 13 that our proposed model can effective classifying these activities by considering activity’s context (location and time).

Fig. 8 HMM recognition precision at different number of states

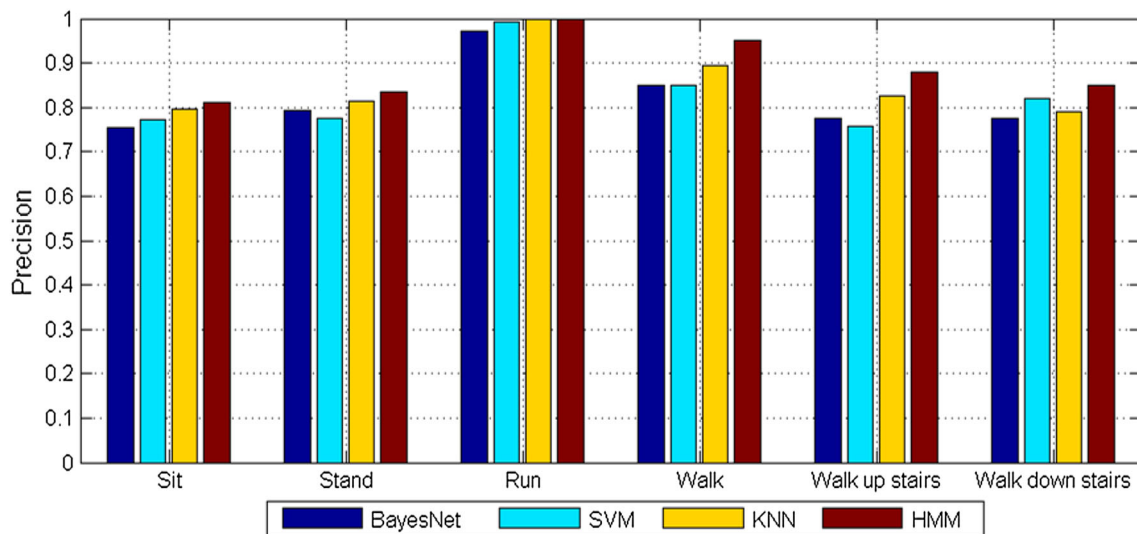
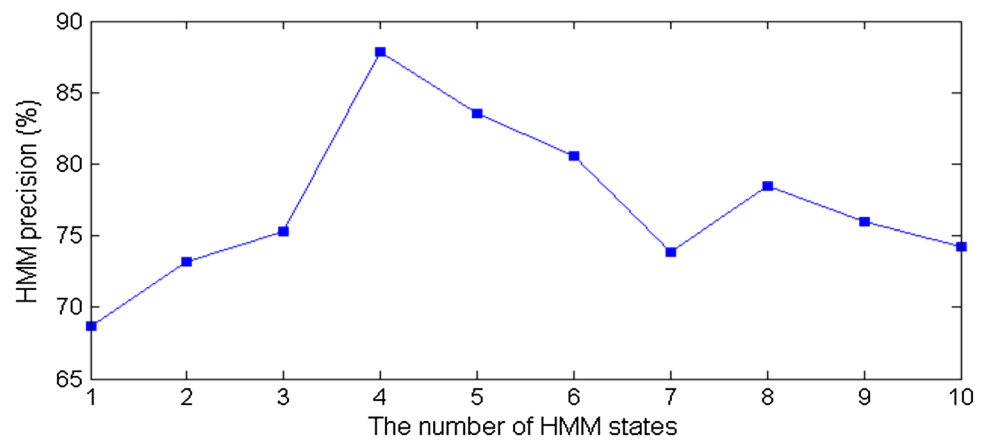


Fig. 9 A comparison of precision for Bayesian network, SVM, KNN, HMM approaches

Table 12 The average precision of activity recognition

Bayesian network	SVM	KNN	HMM	Context-aware HMM
79.8%	85.1%	83.4%	87.8%	92.9%

4.2 Context-aware search system

In order to verify and evaluate whether the search system we proposed is effective, we built a prototype implementation system for our school. As explained before, the system first recognized user's current context based on his/her current activity and other useful information. Note that the activity is recognized by context-aware HMM model based on accelerometer sensor and gyroscope sensor equipped with smart phone. The system retrieves suitable results by context reasoning based on the search content and user's context. We illustrate our technique by means of an example.

Example 1 Suppose that student Aaron wants to search an available classroom for self-study in the next three hours. The detail retrieve process can be seen as follow:

- (1) User's context Recognition based on context-aware HMM.
- (2) Context reasoning considers search content and user context, shown in Fig. 12.
- (3) Retrieve results from context database based on context reasoning.

4.2.1 Experimental setting

We recruited eight participants for our user study and all the participants were PhD students with age between 25 and 30. They performed ten search tasks by using three different systems: our system, a web search system (Google search), and a map search system (AutoNavi map³). In this experiment,

Fig. 10 The precision of context-aware HMM with different values of α

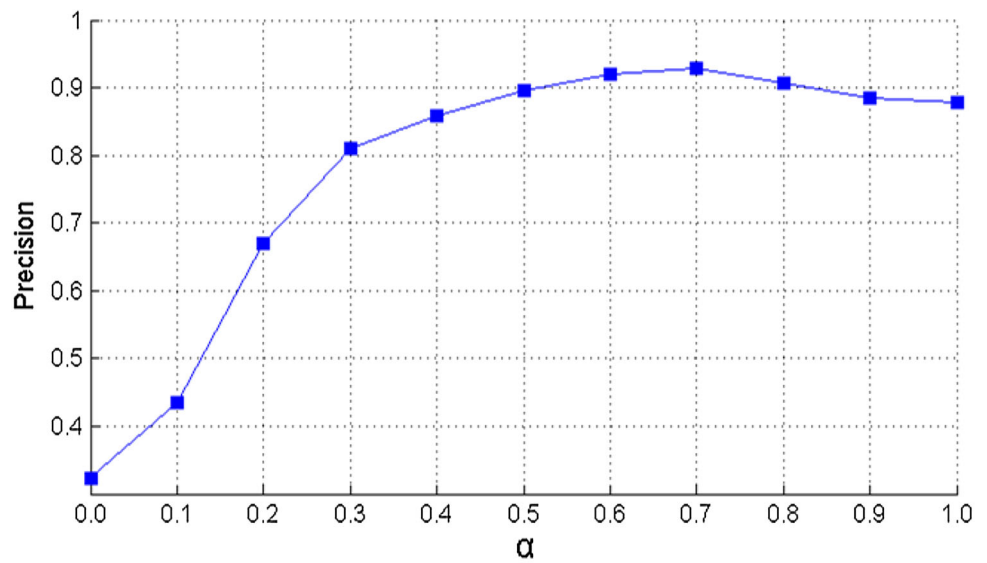


Fig. 11 A comparison of precision between HMM and context-aware HMM

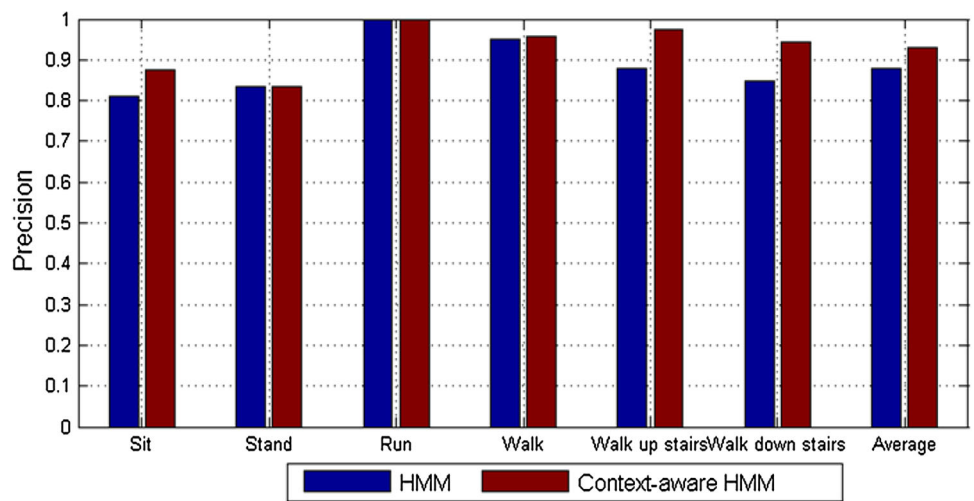


Table 13 Confusion matrix of HMM

Classified as	Sit	Stand	Run	Walk	Walk up stairs	Walk down stairs
Sit	553	94	3	0	0	0
Stand	123	438	11	0	0	0
Run	0	0	615	0	0	0
Walk	0	0	0	544	22	26
Walk up stairs	0	0	0	23	396	45
Walk down stairs	0	0	0	10	20	489

Table 14 Confusion matrix of context-aware HMM

Classified as	Sit	Stand	Run	Walk	Walk up stairs	Walk down stairs
Sit	556	93	0	0	0	1
Stand	78	493	2	0	0	0
Run	0	5	610	0	0	0
Walk	0	0	0	584	3	5
Walk up stairs	0	0	0	11	430	23
Walk down stairs	0	0	0	14	9	497

Fig. 12 Rule-based context reasoning

$$\left\{ \begin{array}{l} (? \textit{person role} : \textit{student}) \\ (? \textit{person situation} : \textit{walking at night}) \\ (? \textit{classroom agenda} : \textit{self - study service}) \Rightarrow (? \textit{classroom provideService} : \textit{self - study}) \\ (? \textit{service time info} : \textit{startTime} > \\ \textit{currentTime} \cap \textit{DuringTime} > \textit{three hours}) \end{array} \right.$$

Table 15 Q1 was the system available to find information you wanted?

Score	-3 (%)	-2 (%)	-1 (%)	0 (%)	1 (%)	2 (%)	3 (%)
Our system	0	2	7	13	27	30	21
Google search	0	4	13	16	35	19	13
AutoNavi map search	12	11	9	19	30	14	5

Table 16 Q2 was the search result suitable for your situation?

Score	-3 (%)	-2 (%)	-1 (%)	0 (%)	1 (%)	2 (%)	3 (%)
Our system	2	0	9	18	32	26	13
Google search	8	13	11	17	27	24	0
AutoNavi map search	0	18	25	31	20	6	0

the participants freely searched for objects in our school in three different situations: morning at laboratory, lunch time at canteen, and walking down stairs at lecture hall. There are nine search tasks in total (i.e., 3 systems \times 3 situations = 9 search tasks). We recorded logs of the operations performed in each system and administered a survey to the participants after they had finishing all tasks. Tables 8 and 9 list the questions in the survey, to which the participants responded with a score from -3 to 3 for each question.

4.2.2 Survey results

In this section, we discuss the results of user studies that we conducted to examine whether our proposed system is effective. As we mentioned previously, we build a context model for our school and we focus on searching content in our school.

Table 15 shows the result of the survey question: Was the system available to find information you wanted? From this result, 78 and 67% of the participants felt that our system and the Web search system, respectively, were convenient for finding information of interest (selected 1, 2 or 3), 49% felt that the map search system was convenient. With a web search system, users can search for all kinds of content on the web. With our system, users can also search for a variety of content, such as location, directions, addresses, but our search system is unable to search information that did not pre-defined in context database compare to web search system. With adding more detailed and new information to context model, our system will perform better. On the other hand,

with a map search system, the kinds of content presented on a map are insufficient for users because they just retrieve information from some predefined and static data. Hence user cannot get current context based information about search.

Table 16 shows the result of the survey question: Was the search result suitable for your situation? From this result, 84 and 63% of the participants felt that the search information was suitable for their context (selected 1, 2 or 3) based on our system and Google search, respectively. Only 44% of the participants felt that the search information was suitable for their context based on AutoNavi map search. Four participants said that our system can get reasonable information to their current context. From this result, while we could confirm to some extent that our system can present useful information considering user's current context, its effectiveness differs among individuals. Therefore, we should consider more situation of user to retrieve more related information.

Figure 13 shows the score of search results based on the three search systems. The score is calculated as follows: for a specific search tasks, each user has a correct answer in his or her mind. For each of the nine search tasks, the user gives the score according to the match extent between the return results and the correct answer provided by users. Table 17 shows the average score for nine search tasks, where a higher score indicates better search results. We can observe that for many participants the searching results of our system were better than the compared systems. In particular, some of the participants indicated that our system significantly outperformed the compared systems when the search task is associated with user's context.

Fig. 13 Score of search results

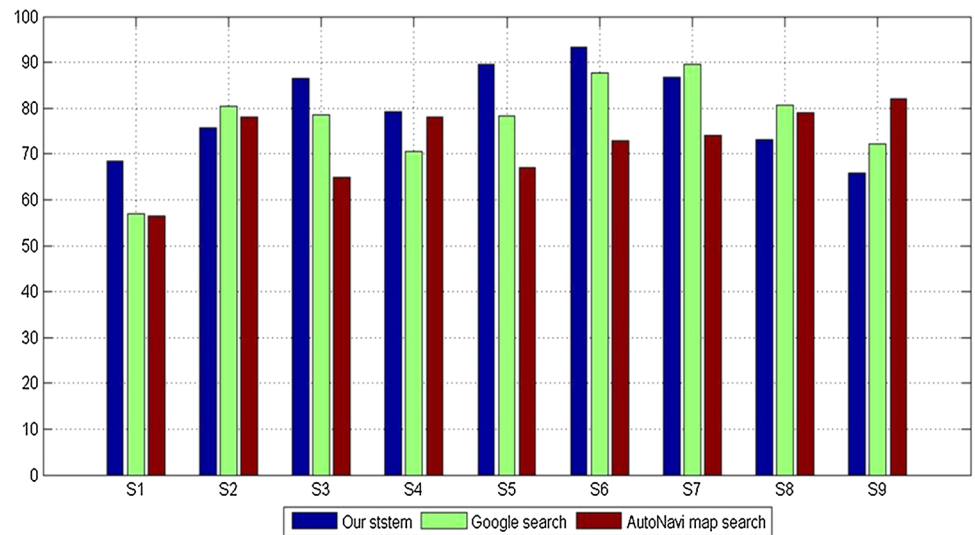


Table 17 Average score of three search systems

	S1	S2	S3	S4	S5	S6	S7	S8	S9
Our system	68	76	87	79	89	93	86	73	66
Google Search	57	81	78	70	78	88	89	80	72
AutoNavi map search	56	78	65	78	67	73	74	79	82

Bold represents better search results

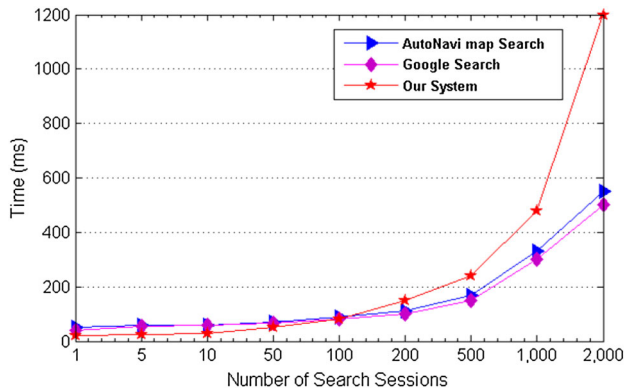


Fig. 14 The running time comparison of the three systems

Moreover, we varied the number of search sessions from 1 to 2000 to record the corresponding running time for comparing the performance of our system and the other two systems. The experiment results are shown in Fig. 14. Obviously, the running time of all three systems increases with the increase of search sessions. It can be seen that the running time of our system is less than the other systems when the search sessions are less than 100, but our system increases sharply when the search sessions are more than 100. The reason for this situation is that our system needs to perform time-consuming context reasoning.

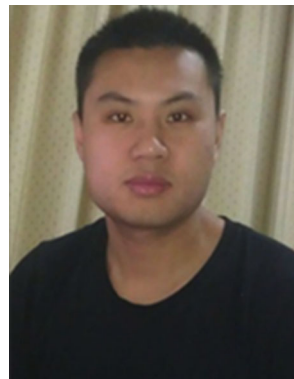
5 Conclusion

In the near future, numerous sensing devices with processing and communication capabilities will be deployed in IoT. Searching information in IoT will become an indispensable application like Internet search. In the paper, we have presented the design and implementation of a context-aware search system for IoT based on hierarchical context model. We use an ontology-based hierarchical context model to represent the IoT objects and their contextual relationships. In our approach, the user’s context is recognized based on a context-aware HMM. Our evaluation shows the superiority of our context recognition model over pure statistical methods. We also conducted user studies to examine the effectiveness of the proposed search system, and confirmed that users could obtain better results than with commercial search systems.

References

- Zhang, D., Yang, L. T., & Huang, H. (2011). Searching in Internet of Things: Vision and challenges. In *Proceedings of the international symposium on parallel and distributed processing and applications* (pp. 201–206).
- Kientz, J. A., Patel, S. N., Tyebkhan, A. Z., Gane, B., Wiley, J., & Abowd, G. D. (2006). Where’s my stuff? Design and evaluation of a mobile system for locating lost items for the visually impaired.

- In *Proceedings of the 8th international conference on computers and accessibility* (pp. 103–110).
3. Kruk, S. R., Grzonkowski, S., Gzella, A., & Cygan, M. (2006). DigiMe-Ubiquitous Search and Browsing for Digital Libraries. In *Proceedings of the 7th international conference on mobile data management* (pp. 84–91).
 4. Frank, C., Bolliger, P., Mattern, F., & Kellerer, W. (2007). Objects calling home: Locating objects using mobile phones. In *Proceedings of pervasive* (pp. 351–368).
 5. Frank, C., Bolliger, P., Mattern, F., & Kellerer, W. (2008). The sensor internet at work: Locating everyday items using mobile phones. *Pervasive and Mobile Computing*, 4(3), 421–447.
 6. Yap, K. K., Srinivasan, V., Motani, M. (2005). MAX: Human-centric search of the physical world. In *Proceedings of the 3rd international conference on embedded networked sensor systems* (pp. 166–179).
 7. Komatsuzaki, M., Tsukada, K., Siio, I., Verronen, P., Luimula, M., & Pieska, S. (2011). IteMinder: Finding items in a room using passive RFID tags and an autonomous robot (poster). In *Proceedings of the UbiComp'11* (pp. 599–600).
 8. Perera, C., Zaslavsky, A., Christen, P., Compton, M., & Georgakopoulos, D. (2013). Context-aware sensor search, selection and ranking model for Internet of Things middleware. In *IEEE 14th International Conference on Mobile Data Management*.
 9. Bergamo, P., & Mazzini, G. (2002). Localization in sensor networks with fading and mobility. In *IEEE PIMRC* (pp. 750–754).
 10. Zhang, D., Zhou, J., Guo, M., & Cao, J. (2011). TASA: Tag-free activity sensing using RFID tag arrays. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 22, 558–570.
 11. Madden, S. R., Franklin, M. J., Hellerstein, J. M., & Hong, W. (2005). TinyDB: An acquisitional query processing system for sensor networks. *ACM Transactions on Database Systems*, 30(1), 122–173.
 12. Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., & Steggle, P. (1999). Towards a better understanding of context and context-awareness. *Handheld and ubiquitous computing* (pp. 304–307). Berlin: Springer.
 13. Wang, H., Tan, C. C., & Li, Q. (2010). Snoogle: A search engine for pervasive environments. *IEEE Transactions on Parallel and Distributed Systems*, 21(8), 1188–1202.
 14. Tan, C. C., Sheng, B., Wang, H., & Li, Q. (2010). Microsearch: A search engine for embedded devices used in pervasive computing. *ACM Transactions on Embedded Computing Systems*, 9(4), 43.
 15. Ejigu, D., Scuturici, M., & Brunie, L. (2007). An ontology-based approach to context modeling and reasoning in pervasive computing. In *PerCom'07* (pp. 14–19).
 16. Hung, P. H., Gu, T., Xue, W., Palmes, P. P., Zhu, J., Ng, W. L., et al. (2009). Context-aware middleware for pervasive elderly homecare. *IEEE Journal on Selected Areas in Communications*, 27(4), 510–524.
 17. Heil, A., Knoll, M., & Weis, T. (2007). The Internet of Things—Context-based Device Federations. In *Proceedings of the 40th annual Hawaii international conference on system sciences* (pp. 58–66).
 18. Ostermaier, B., Roalter, K., Andmer, L. O., Mattern, F., Fahrmaier, M., & Kellerer, W. (2010). A real-time search engine for the web of things. In *Proceedings of the 2nd international conference on the Internet of Things* (pp. 1–8).
 19. Shen, H., Liu, J., Chen, K., Liu, J., & Moyer, S. (2013). SCPS: A social-aware distributed cyber-physical human-centric search engine. *IEEE Transactions on Computers*, 64(2), 518–532.
 20. Hervás, R., Bravo, J., & Fontecha, J. (2010). A context model based on ontological languages, a proposal for information visualization. *Journal of Universal Computer Science*, 16(12), 1539–1555.
 21. Holford, M. E., Khurana, E., Cheung, K.-H., & Gerstein, M. (2010). Using semantic web rules to reason on an ontology of pseudogenes. *Bioinformatics*, 26(12), 71–78.
 22. Maekawa, T., Yanagisawa, Y., Sakurai, Y., Kishino, Y., Kamei, K., & Okadome, T. (2012). Context-aware web search in ubiquitous sensor environments. *ACM Transactions on Internet Technology*, 11(3), 1–23.
 23. Qian, H., Mao, Y., Xiang, W., & Wang, Z. (2010). Recognition of human activities using SVM multi-class classifier. *Pattern Recognition Letters*, 31(2), 100–111.
 24. Khan, A. M., Lee, Y. K., Kim, T. S. (2008). Accelerometer signal-based human activity recognition using augmented autoregressive model coefficients and artificial neural nets. In *Proceedings of 30th annual IEEE international conference on engineering in medicine and biology society* (pp. 5172–5175).
 25. Zhang, D., Zhang, D., Yang, L. T., Gauthier, V., & Xiong, H. (2013). NextCell: Prediction location using cell phone traces. *IEEE Transactions on Computers*, doi:10.1109/TC.2013.223.
 26. Gu, T., Chen, S., Tao, X., & Lu, J. (2010). An unsupervised approach to activity recognition and segmentation based on object-use fingerprints. *Data and Knowledge Engineering*, 69(6), 533–544.
 27. Guan, D., Yuan, W., Lee, Y., Gavrilov, A., & Lee, S. (2007). Activity recognition based on semi-supervised learning. In *Proceedings of 13th IEEE international conference on embedded and real-time computing systems and applications* (pp. 469–475).
 28. Stikic, M., Larlus, D., & Schiele, B. (2009). Multi-graph based semi-supervised learning for activity recognition. In *Proceedings of the 13th IEEE international symposium on wearable computers* (pp. 85–92).
 29. Zhang, T., Liu, S., Xu, C., & Lu, H. (2011). Boosted multi-class semi-supervised learning for human action recognition. *Pattern Recognition*, 44(10–11), 2334–2342.
 30. Baum, L. E., Petrie, T., Soules, G., et al. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. *The Annals of Mathematical Statistics*, 41, 164–171.
 31. Zhang, D., Yang, L. T., Chen, M., Zhao, S., Guo, M., & Zhang, Y. (2014). A real-time locating system using active RFID for Internet of Things. *IEEE Systems Journal*, doi:10.1109/JSYST.2014.2346625.
 32. Zhang, D., Guo, M., Zhou, J., Kang, D., & Cao, J. (2010). Context reasoning using extended evidence theory in pervasive computing environments. *Future Generation Computer Systems (FGCS)*, 26, 207–216.
 33. Husne, A. R., & Tae-Seong, K. (2010). Activity recognition from single tri-axial accelerometer using wavelet packet decomposition and ANN. In *Proceedings of the 2010 international conference on artificial intelligence* (pp. 386–390).



Yuanyi Chen is a Ph.D. candidate at Shanghai Jiao Tong University. His research topic includes Internet of Things, mobile computing, social networks and wireless sensor networks.



Jingyu Zhou received the B.Sc. degree in computer science from Zhejiang University, China, in 1999. He received the M.S. and Ph.D. degrees in computer science from the University of California at Santa Barbara in 2003 and 2006, respectively. He joined the Software Institute at Shanghai Jiao Tong University in 2006. He is generally interested in information retrieval, systems, and security. His current work includes vertical web search, including information retrieval, Chinese analysis, and search systems. His past work

includes network and application security, parallel scientific computing, cluster-based storage systems and middleware systems, and cluster-based Internet services.



Minyi Guo received the B.S. and M.E. degrees in Computer Science from Nanjing University, China in 1982 and 1986, respectively. From 1986 to 1994, he had been an assistant professor of the Department of Computer Science at Nanjing University. He received the Ph.D. degree in information science from University of Tsukuba, Japan in 1998. His research interests include parallel computing, distributed computing and mobile computing. He is an associate editor for IEEE Transactions on Parallel and Distributed, and IEEE Transactions on Computers.

Transactions on Computers.