Pers Ubiquit Comput (2018) 22:201–218
https://doi.org/10.1007/s00779-017-1087-0

ORIGINAL ARTICLE

Personal search system based on android using lifelog and machine learning

Yunjin Nam[1] · Dongkyoo Shin[1] · Dongil Shin[1]

Received: 25 September 2017 / Accepted: 26 October 2017 / Published online: 7 November 2017

Abstract Lifelog is the foundation on which lifelong services and healthcare services are implemented in a smart home system. It also plays a major role in the sub-processes of the system because it acquires information about the home's residents for home automation and entertainment. Providing personalized services to individuals by acquiring and managing this personal lifelog information has great advantages in terms of service satisfaction and effectiveness. In this paper, we implemented a personal search system based on android that collected and stored an individual's lifelog based on nine smart phone sensors and used it to derive new meaningful information about the user. The activity recognition module for classifying the user's behavior, the naive Bayesian method, showed an accuracy of 88.23% and the area under the ROC curve value of 0.941. We designed and implemented density-based spatial clustering method in the module for extracting the point of interest and the participants filled out a satisfaction questionnaire to evaluate the search system. The proposed system efficiently uses a large amount of lifelog data and automates the process of extracting meaningful information, associating it according to the user's intention.

Keywords Lifelog · Mobile · Activity classification · Naive Bayesian · Point of interest · User feedback

✉ Dongkyoo Shin
shindk@sejong.ac.kr

Yunjin Nam
namyunjin0912@gmail.com

Dongil Shin
dshin@sejong.ac.kr

[1] Department of Computer Engineering, Sejong University, 98 Gunja-Dong, Gwangjin-Gu, Seoul 143–747, South Korea

1 Introduction

With the development of mobile devices and the Internet environment, everyone can use devices at any time anywhere. According to the National Statistical Office's 2016 estimated population and Nielsen Koreanclick's population estimation survey report, the proportion of mobile devices is increasing with the passage of time. The media device usage proportions for those born from 1981 to 1995, which accounted for 21% of the population in 2017, were 95.3%, and those born after 1995, which accounted for 15% of the population, had a rate of 92.7% [1]. In other words, over time, the usage rate is increasing as the generations change, and mobile devices exist in most parts of the society and in the daily lives of users.

As the form and function of mobile devices have rapidly developed, they have provided users with great convenience, and mobile devices have become a necessity for members of society. The introduction of functions, contents, and services for the elderly and children, as well as those who are actively engaged in economic activities, has lowered the barriers to access mobile devices and made it easier [2].

Members of modern society always carry their mobile devices or smart phones in their pockets or bags and use them at anytime and anywhere. The log of an individual user, such as a log that reveals the characteristics of the individual possessing the smart phone, is referred to as their personal lifelog. Because the convenience and portability of smart phones are much better than those of other devices, the number of lifelogs that are used and collected by many users is rapidly increasing. Logs of personal characteristics can be fed back to the user more effectively if they are processed and analyzed by the management system [3, 4]. Because of this feature, a lifelog is used as the basis for lifelong services and healthcare services, which can be implemented in a smart home system. It also plays a major role in the sub-processes of the system because it acquires information about a home's residents for home entertainment.

A lifelog system that only collects and stores logs cannot define the information to be fed back, and the feedback function will be a simple log search. Therefore, the satisfaction that can be expected in providing user feedback is enhanced by not only storing the generated logs but also effectively generating additional meaningful information and providing information related to the desired intention or function.

In this paper, we propose a personal search system that can consider the user's intention to provide efficient search results. In order to build a base for searching, we have defined and fixed the collected logs and added information to the database of the server. We implemented a module that extracts a user's point of interest about a location and the user's activity information from the system. We also propose a personal search system that constructs a network that can relate the data using stored data in the information processing process and can feed back the results in accordance with the user's intention along with the inferred results from the system.

In Section 2, we discuss the concept and related research of lifelog and the semantic network in relation to the proposed system. Section 3 explains the types of collected data, the process of data collection and storage, meaningful information extraction for effective feedback and configuration, and retrieval method of data connection network, and develops the chapter according to the overall flow of the personal search system. In Section 4, accuracy of activity recognition module and the user satisfaction experiment are conducted in order to examine the application and results of the lifelog system. Section 5 concludes this paper.

2 Related work

2.1 Lifelog

Logs are generated according to events that occur naturally using devices with mobile characteristics, with information recorded in the device or a record of daily life according to the user's needs. Widely used devices such as those for the Internet, applications, watching movies, and listening to music take up a large part of a modern person's life and generate large amounts of logs and information. The logs that often arise from an individual's daily life or experience are called lifelogs.

The lifelog system is the first to acquire and store data through a PC or server that can process data acquired from arbitrary users and various devices. The process of data processing, analysis, and feedback to the user acts as a basic system flow, and the basic structure of the system is shown in Fig. 1.

SenseCam, which records images of everyday life, was used to implement a PC-based program that manages the daily activities of users or to derive the types of human activities using visual information [5–8]. In addition, research has been conducted on a lifelog monitoring system that models the walking behavior for people of specific ages and logs audio information through a wearable device, indexing and annotating the information to facilitate auditory information research [9, 10]. However, in these studies, the users' daily life information had to be obtained using composite elements such as wearable devices or sensors located in the experimental environment.

Lifelog plays a key role in a smart home environment because it learns about a home's residents for home entertainment and plays a major role in the sub-processes. In particular, a resident's lifelog data can be used to improve their quality of life by identifying their behavior pattern in daily life and monitoring it continuously. In this regard, research on recording and recognizing human activities by implementing a real-time lifelog system using a deep imaging device and analyzing the context of large-capacity data using a wireless sensor network has been actively carried out [11, 12].

Mobile devices are advantageous in that they can continuously obtain various information from various sensors in the device. The types of sensors include an acceleration sensor, a gyroscope sensor, and a GPS, which are event values that occur according to a user's motion. The available information includes telephone calls, messages, and time data, which can be obtained using the basic functions of a mobile phone. Figure 2 shows the types of basic information collected from mobile devices.

Much research has been done on methods for tagging or logging one sensor by fusing sensors to calculate information. For example, studies have been conducted to automatically index streams of everyday experiences collected using multiple sensors on mobile devices and to detect and store a user's context information through continuous sensor detection to help record personal daily events [13].

Mobile devices dominate the telecommunications market and are used by many people. The various sensors built into the device make it easy to gather information about their natural daily life, which is difficult to collect using sensors mounted in a room or on the body. Devices can acquire much information, but the indiscriminate collection of information can cause system overload or unnecessary operations. For this reason, it is necessary to define the sensors and sensor values to be collected according to the purpose of the system and to organically link the defined contents in order to avoid providing unrelated feedback results.

2.2 Semantic network

The term semantic network represents the structure of a network that can communicate with people. It can be called a

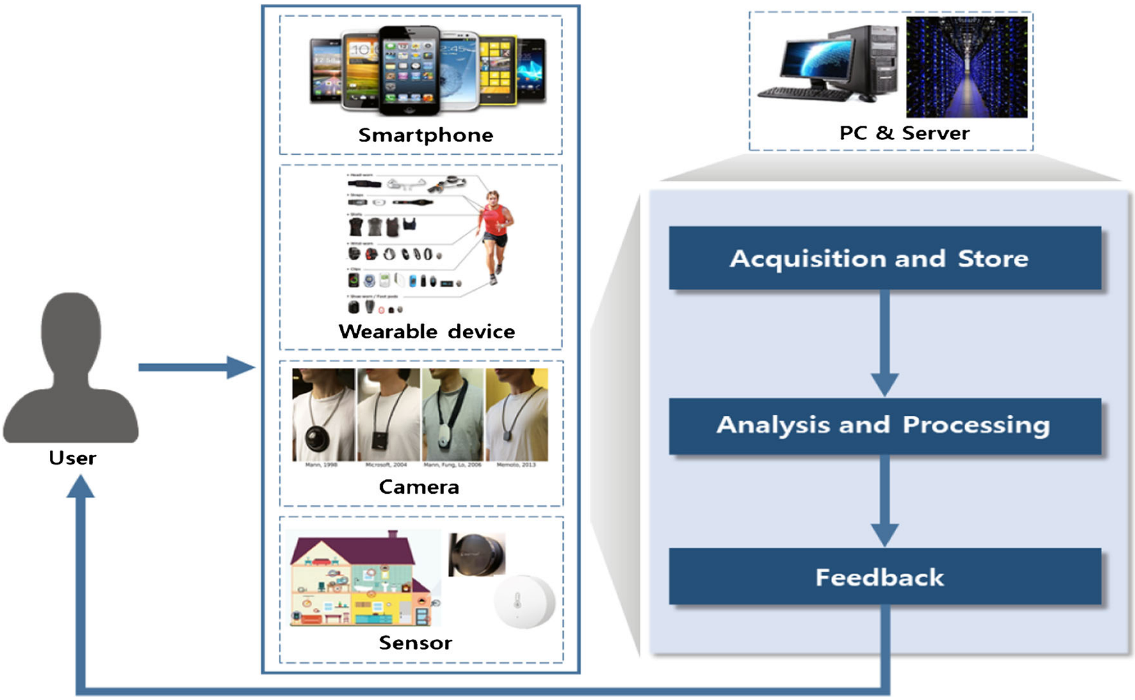


Fig. 1 Basic structure of lifelog management system

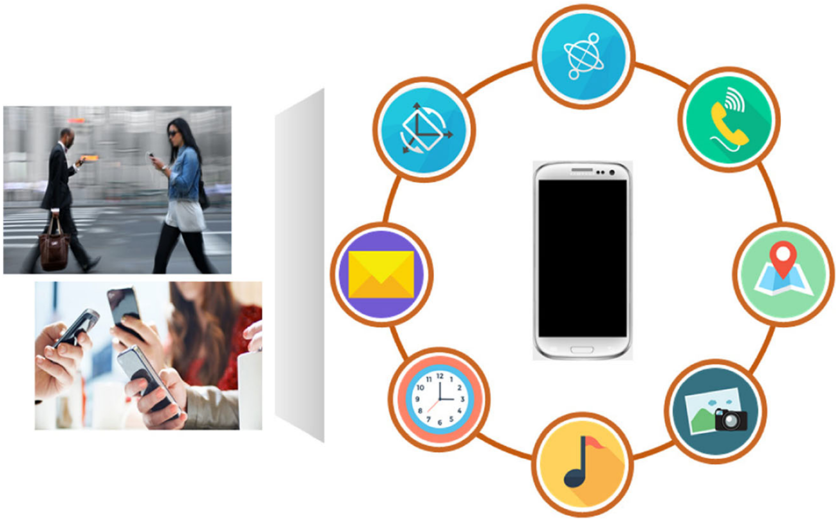
Semantic Web because it is mainly used to develop search technology using information on the Web [14, 15].

It maps the information defined by a person to nodes and edges in the network structure and recognizes the meaning of the system. It grasps a user's query and searches and retrieves the network structure to provide the necessary information or related information.

Since a general keyword-based search is performed through a process such as morphological analysis, it is impossible to search when the keywords in the search query are not the same as the keywords in the information to be searched. On the other hand, a semantic network has the advantage of being able to search using semantic reasoning because it has a structure based on mutual relations based on meaning. Because it is possible to search by the relations between keywords, it is possible to provide additional association information through judgments of the network, including a search similar to a general search.

The related studies were mainly conducted on the Web. Thus, Web users could provide effective feedback to obtain information. For example, research was conducted on an early Semantic Web search that added annotations and markups to the Web, searched by Crawler, and conducted relation formation generation between multimedia resources using a

Fig. 2 Basic information gathered from mobile devices



semantic link network model in relation to big data, which is a trend of our information society [16, 17].

The basic structure of the semantic network refers to Tim's Semantic Web technology structure [18, 19]. The components in Fig. 3 include URIs and Unicode for languages that represent resources. These include XML (Extensible Markup Language), RDF (Resource Description Framework), RDF schemas, ontologies, and RIF (Rule Interchange Format) for a hierarchy and contextual representation hierarchy. OWL supports semantic functions and reasoning, and RIF differs from query and rule expressions. In addition, we demonstrated a conclusion hypothesis derived from logic, which draws new conclusions by means of logic laws. Trust ensures that the source of information from the web is trusted and the interfaces and applications for the user are located.

In particular, the part related to this study is the part where XML and RDF are the main part: XML provides the structural definition between documents and objects, and RDF is the basis for expressing it as XML metadata. The semantics of the terms that cannot be defined by XML and RDF alone are provided and specified so that the computer itself can recognize the meaning of the document or object.

Figure 4 shows an example of applying the semantic network structure to animals and objects. It can be noted that the nodes and edges are connected by the necessary semantic relations. The data structure connected to the network structure is mainly used for social log analysis or searching the web because there is an advantage in performing a search.

3 Design and implementation of the lifelog system

In this study, we extracted meaningful information from a user's lifelog, used it as a new data attribute, and conducted a search using the attributes of data based on the semantic network structure. Unlike a general search method that searches all the indexes or data or compares only keywords, the related search result value that includes helpful information is transmitted by performing a process of comparing the values of the categories to which the keyword belongs.

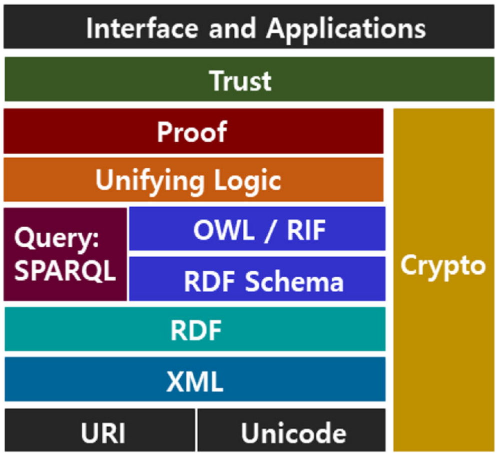


Fig. 3 Basic structure of semantic network (Semantic Web)

Figure 5 shows the structure and flow of the whole system designed and implemented in this study. The log input from the mobile user and the meaningful information extracted from the acquired log are integrated into the database of the lifelog management system. The data in the integrated database include data attributes, data types, and user actions, and the relationships between these data are defined hierarchically, as illustrated in Section 3.4.

The categories and associations are defined by dividing the activities related to the user's daily life into five categories and are used based on the network structure by staging the relationships between the data.

When the user wants to search for an episode that occurred to them, they input the condition and question through a query. When a query is entered, the system maps the query keyword to the defined step attribute so that only the related information can be retrieved through the attribute.

If the retrieved association information and results of the question can be derived from the system, the relevant results can be fed back to the mobile device, and the analysis results of the lifelog data itself can be browsed if the user desires.

3.1 Data structure of the proposed lifelog system

In this study, we designed and implemented a lifelog system that can perform an associative search using a mobile device and PC. It enables the personalization of the modeling process from the lifelog through the interaction of the Android client system, the server system, including an environment such as PHP, and the database. The configured models and networks are effective in providing feedback tailored to the user's intent.

The data acquired from Android is usually non-formalized. Thus, it should prevent the collection of indiscriminate information. Therefore, it is necessary to define the proper data to be input to the lifelog system, which can effectively show the user's personal daily life.

In this study, we implemented the system for Android, in which we retrieved the values of various sensors inside the mobile device. Table 1 defines the types and properties of the sensors used in the research. The system collected information about calls, SMSs, and pictures, which were contained in the logs related to actions that the user performed or intended to perform. Accelerators, gyroscopes, and GPS sensors were used to collect information about events.

The Android API 23 and later policies ask for permission and non-permission to read and write data. We started collecting only the user permissions and could select the type of data collected at any time, depending on the application settings or when the user started the module. Figure 6 shows a flow diagram of the module for collecting and storing logs.

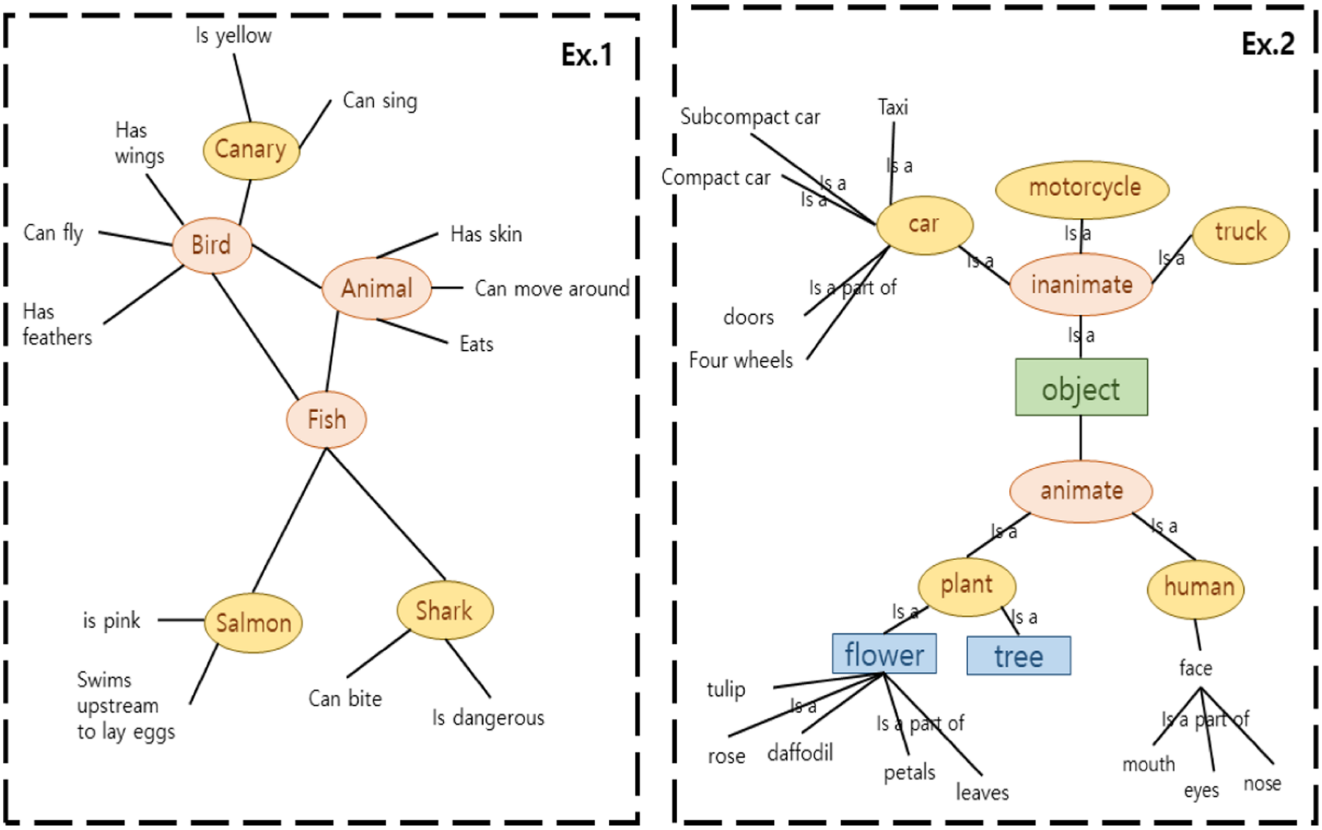


Fig. 4 Example of applying semantic network

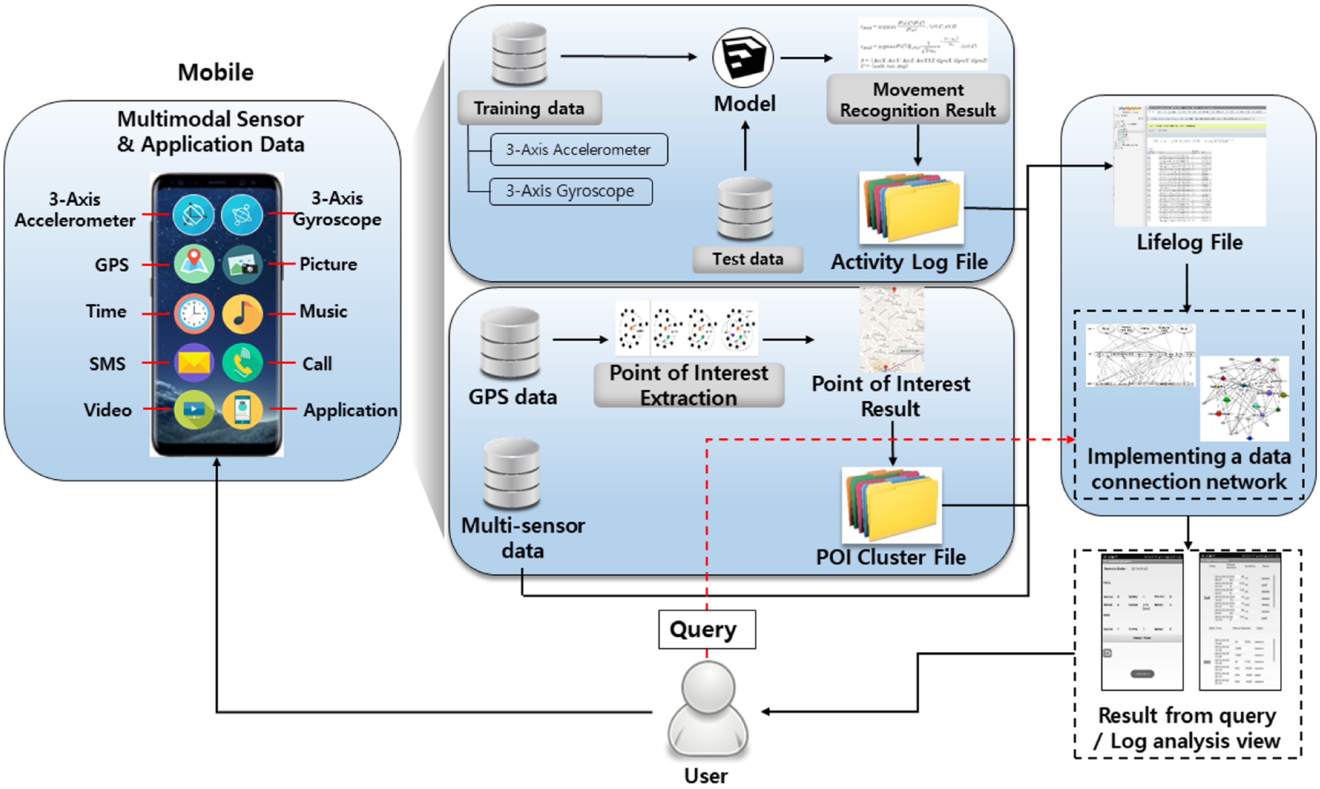


Fig. 5 Structure of lifelog management and search system

Table 1 Types of collected data

Type	Attributes
Acceleration	*X*-axis value, *Y*-axis value, *Z*-axis value, SVM (signal vector magnitude), time
Gyroscope	*X*-axis value, *Y*-axis value, *Z*-axis value, time
SMS	Number, text, type, time
Call	Number, duration, type, time
GPS	Longitude, latitude, time
Picture	Title, format, longitude, latitude, context type, time
Music	Title, metadata, time (start, end), duration (play)
Video	Title, metadata, time (start, end), duration (play)
Application	Title, type, time (start, end)

3.2 Information generation—activity recognition

The log gathered was not sufficient to obtain a fully personalized profiling file. The information that is fed back through the personal search system has a greater effect on user satisfaction and service when more personal characteristics are included. In the system implemented in this study, additional information is generated using basic data acquired through the lifelog collection and storage module for effective feedback, and it extracts meaningful information to help create a lifelog base file for the system. The information generation module for effective feedback includes an activity classification module and a point of interest extraction module.

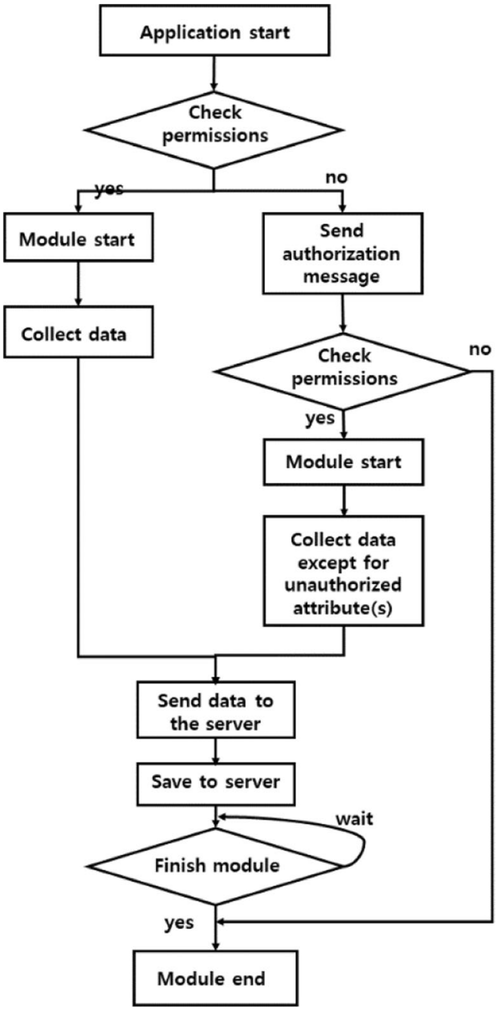


Fig. 6 Flow chart of lifelog collection and storage module

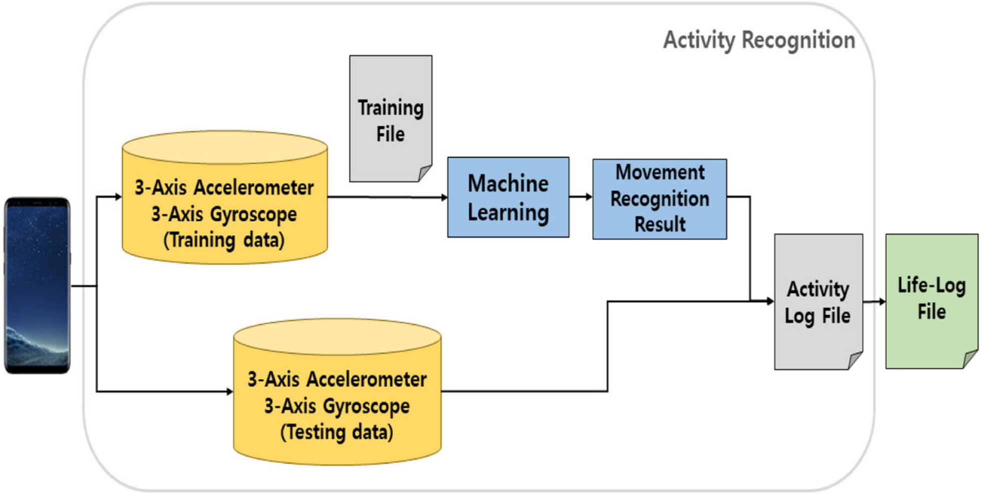
The activity classification module provides an intelligent classification system by applying a machine learning algorithm to the system rather than classifying the sensor values using the existing thresholds. It is designed to create a model suitable for an individual person and perform activity classification without receiving a threshold value directly from a user or specifying an appropriate threshold value.

The basic flow is as follows. After applying the machine learning algorithm from the training file, a model file is created and saved. The information extraction module for the activity classification receives the training file according to the user's intention. Activities related to walking, running, and stopping are input from the user through the button, and the activity related to the user is performed for 20 s to produce an action training data set for the user. If the user does not have write/read permission, the module can be terminated immediately because it cannot create the necessary file for activity classification.

The acceleration sensor and gyro sensor values are generated every 10 s by the user's movement and are received in real time and sent to the server. Figure 7 shows the overall structure of the activity classification module, and Fig. 8 is a flow chart showing the steps until the training file is stored in the server and the model is created.

In addition to the values for the acceleration sensors (AccX, AccY, AccZ) and gyro sensors (GyroX, GyroY, GyroZ), the AccXYZ (signal vector magnitude) value is available in the training data set. It is an attribute added to provide a single representative value. Equation (1) gives the signal vector magnitude used to process the *X*, *Y*, and *Z* values output by the triaxial accelerometer into one representative value, for

Fig. 7 Overall structure of activity classification module



example, the one which will be used in the application for tracking walking speed:

$$\mathrm{AccXYZ} = \sqrt{(AccX)^2 + (AccY)^2 + (AccZ)^2} \tag{1}$$

In order to select the algorithm to be used in the activity classification module, we evaluated the efficiency of two algorithms. The target algorithms were naive Bayesian, a probabilistic classification algorithm, and SVM (support vector machine), a statistical classification algorithm. Based on the experimental results, the naive Bayesian method was applied to the system.

3.3 Information generation—point of interest extraction module

The point of interest (POI) extraction module extracts the user's POI using GPS. The POI is used to determine the location of a place that is meaningful to the user, considering the place where the user is staying, or staying for a long time. Through the GPS acquisition part of the module, the server receives the location once per minute from the user and performs location extraction for the first-order data that can best represent the user's activity pattern. The density-based spatial clustering of applications with noise (DBSCAN) is used.

The DBSCAN is an algorithm that automatically generates clusters based on a density calculation by specifying a minimum number of points satisfying epsilon and clusters, unlike the existing *K*-mean algorithm. In order to avoid providing unreliable clusters as a result of a small number of clusters or large amount of indiscriminate generation, we applied an algorithm that can create clusters based on data.

In this system, a condition for creating a cluster is specified, and when a cluster is created, a point of interest is determined through the process of calculating the center point of the generated cluster. In calculating the center point, the center point of each cluster is first calculated. Then, another center point between the center points is calculated and designated. Even if the user does not pass the specified center point, a location near the location or a location related to the user's experience is designated because it is a point calculated by the information passed by the user. This will be helpful when providing location-based services or searches to users.

Figure 9 is a diagram showing the application of DBSCAN, and Fig. 10 is the pseudo-code for the cluster configuration algorithm operating on the system. Figure 11 shows the application screen before and after applying the cluster algorithm. It can find the information of the corresponding position through the coordinates of the point of interest and API.

Figure 9 is a diagram showing the application of DBSCAN, and Fig. 10 is the pseudo-code for the cluster configuration algorithm operating on the system. The *applyDBSCAN()* in Fig. 10 shows the process of searching all the points belonging to the log in order to find clusters satisfying the number of minimum points specified by the system. The process of adding the point satisfying the condition to the cluster is performed. The condition satisfies the radius (epsilon, 0.0001), and the *getNeighbors (p)* is performed to find it.

Figure 11 shows the application screen before and after applying the cluster algorithm about two files. The figure on the left of the arrow shows the map of the raw GPS log data before applying the DBSCAN. On the other hand, the figure on the right shows the cluster center point as a marker after clustering. The center point of the cluster is specified as the closest point to the middle point among all the points constituting the cluster.

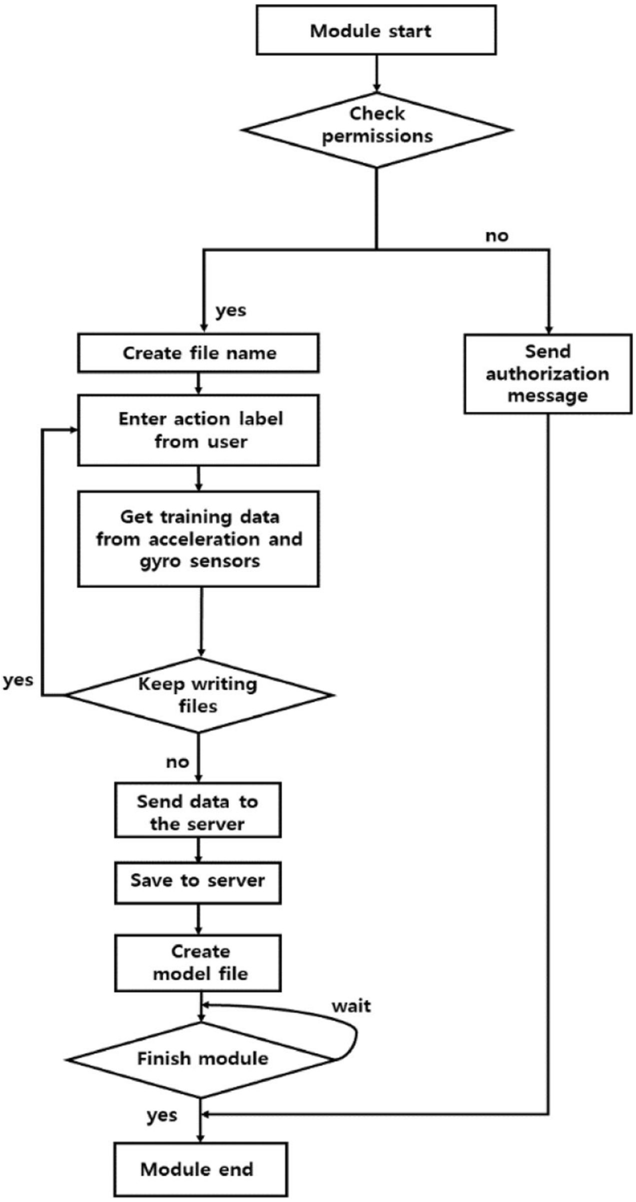


Fig. 8 Flow chart of activity classification module

It can find the information of the corresponding position through the coordinates of the point of interest and the API.

3.4 Semantic discovery by configuring data connection network

By applying the data connection network structure based on the semantic network structure to the lifelog data, we can connect the related data and make the association search possible. Based on this method, it is possible to create an episode using lifelog beyond the related information retrieval shown in this paper and to formulate data that are collected indiscriminately.

To apply the structure, it is necessary to define the correlation of the defined lifelog data. By defining the correlation, information about the attributes and values of the data can be expressed by nodes and edges when learning the network structure. Figure 12 defines the mapping of the attributes and values of the data used in this system to the network structure.

To create an RDF data model based on Fig. 11, an informal sample triple is created in Table 2. A triple is a relation that has a direction from a subject to an object (property), and it is a sentence having a structure expressing the characteristics of a subject and an object.

Based on the RDF triples shown in the example in Table 2, the subject and object can be represented as one node, with the attribute as an edge. Figure 12 shows the result of applying the network structure using the graphic tool provided by NetBeans for the data attributes collected based on the definitions and Table 2. Figure 13 shows the network structure of the overall semantic structure that is connected through layer 1, layer 2, and layer 3.

The semantic-based retrieval method proceeds in the same way as the example shown in Figs. 14, 15, 16 and 17. The query used in the example is a search for "Where is the place at the time of listening to certain music?" It is assumed that a search query of "I want to find the 'place information' when I heard 'Closer' from 'The Chainsmokers'" is executed.

Music is expressed as a set using formula notation (2), where the title is "Closer" through the user's search query, and "The Chainsmokers" is the artist part of the metadata that is mapped. Putting the mapped set on the search system allows the system to search for the start time.

$$\text{Music} = \{\text{title}, \text{metadata}, \text{start time}, \text{end time}, \text{duration}, \text{type}\} \tag{2}$$

In other words, because the search system should show the search result according to the condition of "When," the search for the attribute of layer 1 is associated with music and the related layer 3 becomes the primary. Figure 14 shows the relevant contents.

When the primary search is completed, the secondary search is performed with respect to the start time. Because the user's ultimate goal is location information, the system needs to find information on the "POI" that is linked through layer 2 and layer 3. However, because the start time and POI belong to layer 1, only layer 2 should be searched. Therefore, at the end of the primary search, it is possible to provide information on the overlapped portion in comparison with layer 2 associated with the "start time," and layer 2 is connected with respect to the "POI." Figures 15 and 16 show the relevant information.

In the secondary search, the layer result associated with the condition information provided by the user query and the layer result associated with the final target result node are

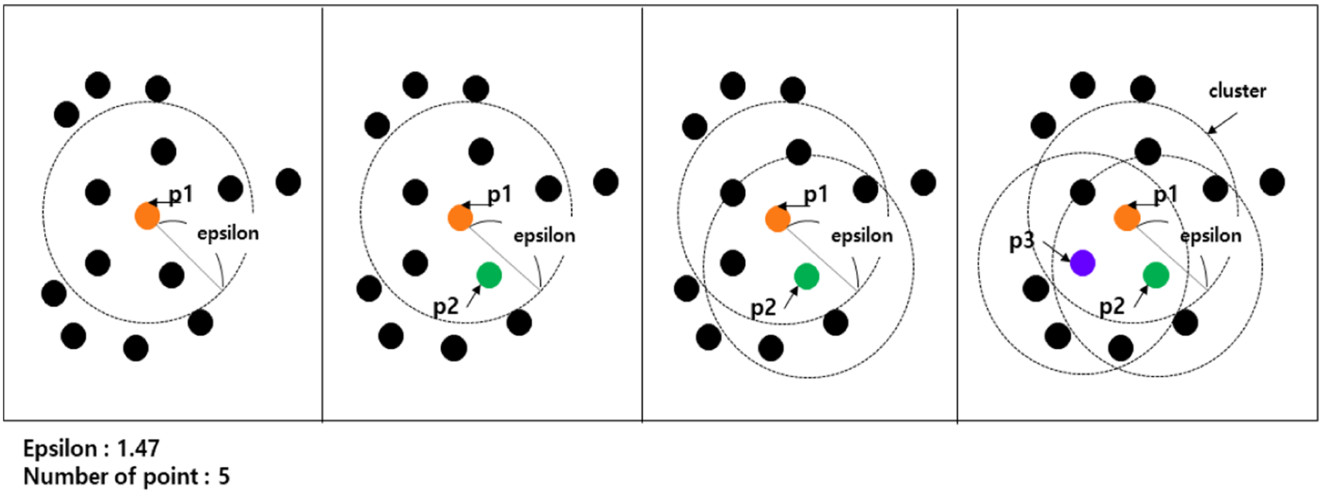


Fig. 9 DBSCAN development process

searched together. If you compare the results according to the scenario, it appears as $\text{Search}_{\text{start time}}$ = (Internet, music, call, video, application, picture, SMS, GPS), $\text{Search}_{\text{POI}}$ = (picture, GPS). Because the overlapping categories are GPS and picture, the system can provide related information according to the user's intention through these two categories.

```
def applyDBSCAN():
 while pointList Size > index: /*index is 0*/
   p = pointListGet(index)
   if p is not visited:
     visited(p)
     Neighbors = getNeighbors(p)
     if Neighbors Size > min point: /*min point is 10*/
       ind = 0
       while Neighbors Size > ind:
         r = Neighbors Get(ind)
         if r is not visited:
           visited(r)
           SecondNeighbors = getNeighbors(r)
           if SecondNeighbors Size >= min point: /*min point is 10*/
             Neighbors = Merge(Neighbors, SecondNeighbors)
         ind++
       resultList Add(Neighbors)
     index ++
   return resultList


def getNeighbors(p):
 pointList = getList
 neighbor = new ArrayList
 points = pointList iterator()
 while points has Next:
   q = points next()
   if getDistance(p, q) <= epsilon /*epsilon is 0.0001, 10m*/
     neighbor Add(q)
 return neighbor
```

Fig. 10 Pseudo-code for cluster configuration using DBSCAN

After the secondary search, a result selection process is needed to provide the user with the desired results. In the tertiary search, a selection is made on the results to be fed back. The system can find the longitude and latitude corresponding to the start time under the condition entered by the user and select and provide the point of interest.

Figure 17 shows the nodes related to GPS, which are represented by formula notation (3).

$$\text{GPS} = \{\text{start time}, \text{point of interest}, \text{longitude}, \text{latitude}\} \quad (3)$$

For the GPS located at layer 2, because the start time (event time) is recorded, it is possible to retrieve the latitude and longitude coordinates corresponding to the time of the condition obtained by the secondary search. The center of the cluster, which includes the location or the latitude and longitude, is transmitted to the user through the early process of the t search, in which the latitude and longitude are searched. The system also checks for relevant photo records and if there is a log of a photo at that time, it can be delivered together with the results that matched the first intended question.

4 Experiments and comparison analysis

In this section, we compare the results of machine learning algorithms to select the algorithms to apply in the design and implementation of the activity recognition module. We also discuss a survey that was conducted to determine the user satisfaction with the overall system and describe the results.

The server computer running the personal search system was an AMD FX-8370 Eight-Core Processor, 32.0 GB memory, 64-bit Windows 10 environment. Experiments were conducted with five individuals who privately owned Android

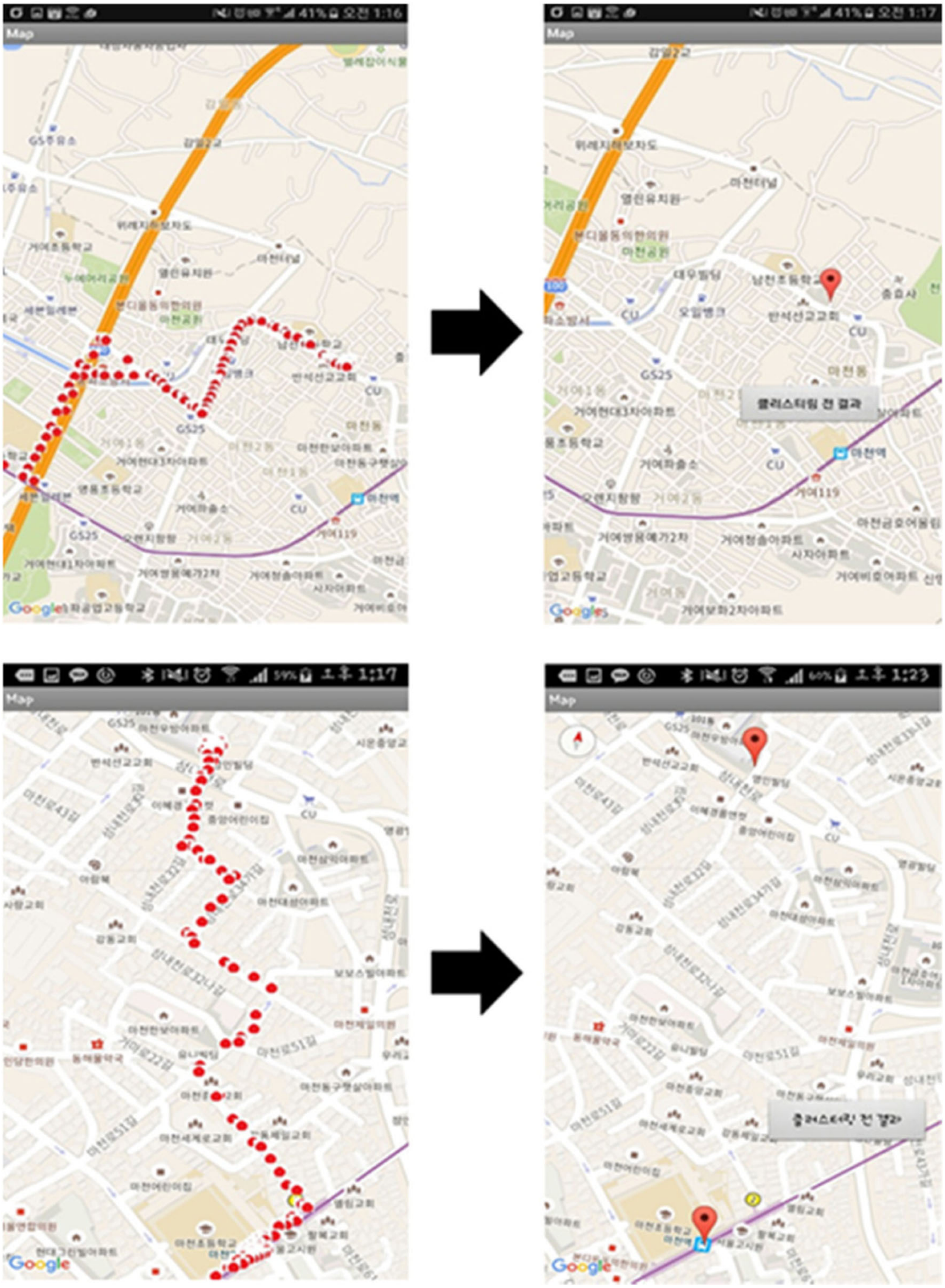


Fig. 11 Smart phone screen before and after applying cluster configuration algorithm

mobile devices. In the feedback experiment, for each participant, three questions were put as input, with a total of 15 questions to input and process, and the user satisfaction is described in Section 4.2.

4.1 Experimentation of the activity recognition algorithm

In this section, we discuss how the performance and efficiency were evaluated by conducting experiments on two algorithms,

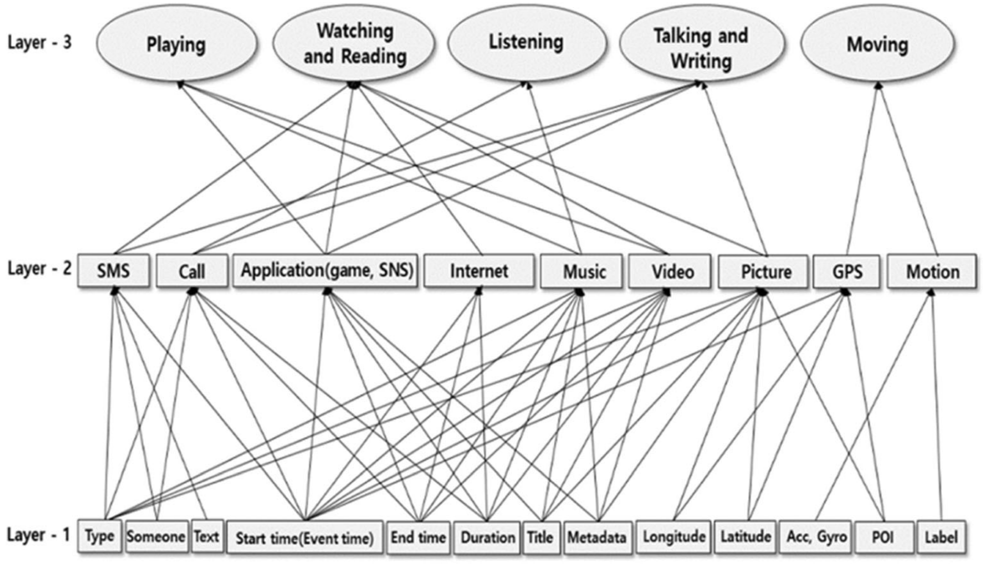


Fig. 12 Mapping of the attributes and values of the data used

the naive Bayesian and SVM, based on the basic structure and flow of Section 3.2 to select the algorithm to be used in the activity classification module.

The seven attributes (AccX, AccY, AccZ, GyroX, GyroY, GyroZ, AccXYZ) could be obtained and used from the acceleration sensor and gyro sensor. It is composed of three axis values in the acceleration sensor (AccX, AccY, AccZ), three axis values in the gyro sensor (GyroX, GyroY, GyroZ), and a representative value (AccXYZ) that does not take into account the directional properties. All property values had to be processed using probability distributions because they had continuous real or integer types.

The first applied algorithm was naive Bayesian, which was used to derive the prior probability from the training data file, derive the posterior probability for the new value, and classify it with the highest probability. At this time, the probability distribution used the Gaussian distribution to apply the algorithm.

Table 2 RDF Sample Triples

Subject	Property	Object
Music	Has	Metadata
Picture	Has	Metadata
SMS	Was written at	Time
Call	Occurred at	Time
Listening	Has	Call
Listening	Has	Music
Application	Was run during	Duration
Video	Was run during	Duration
Music	Was run during	duration

Equation (4) is the basic naive Bayesian application concept, where C (c) is the class (activity), and S (s) is the sensor (attribute). Equation (5) is a formula calculated after applying the Gaussian distribution. μ_{m} represents the average of μ_1, …, μ_{n}, which means the average, and μ_{v} represents the average of $\sigma_1{}^2$, …, $\sigma_{\mathrm{n}}{}^2$, which represents the variance. Equation (6) is a formula notation for a set of classes and sensors corresponding to Eqs. (4) and (5). Equations (4), (5), and (6) are part of the activity recognition algorithm and are located in machine learning in Fig. 7.

$$C_{MAX} = argmax\frac{P(s|C)P(C)}{P(s)}, (c\in C, s\in S) \quad (4)$$

$$C_{MAX} = argmaxP(C)\prod_{s\in S}\frac{1}{\sqrt{2\pi\mu_{\mathrm{v}}}}e^{-\frac{(v-\mu_{\mathrm{m}})^2}{2\mu_{\mathrm{v}}}}, (c\in C, s\in S) \quad (5)$$

$$S = \{AccX, AccY, AccZ, GyroX, GyroY, GyroZ, AccXYZ\}, C = \{walk, \mathrm{run}, stop\} \quad (6)$$

The system was implemented according to the equation, and the classifier performance experiment was performed for each behavior. Table 3 summarizes the results for the AUC (area under the receiver operating characteristic curve), which indicates that the accuracy is closer to 1 and the classification algorithm is better. TP (true positive rate) is the ratio of the case where the actual correct answer (positive) matches the experimental result (positive), and FP (false positive) represents the ratio of the difference between the actual answer (negative) and the experimental result (positive). Precision is actually a proportion of the right results out of the right-categorized experiments, and recall is the percentage of experimental results that are correctly classified during the actual answer.

Fig. 13 Semantic network structure of data attributes by relationship definition

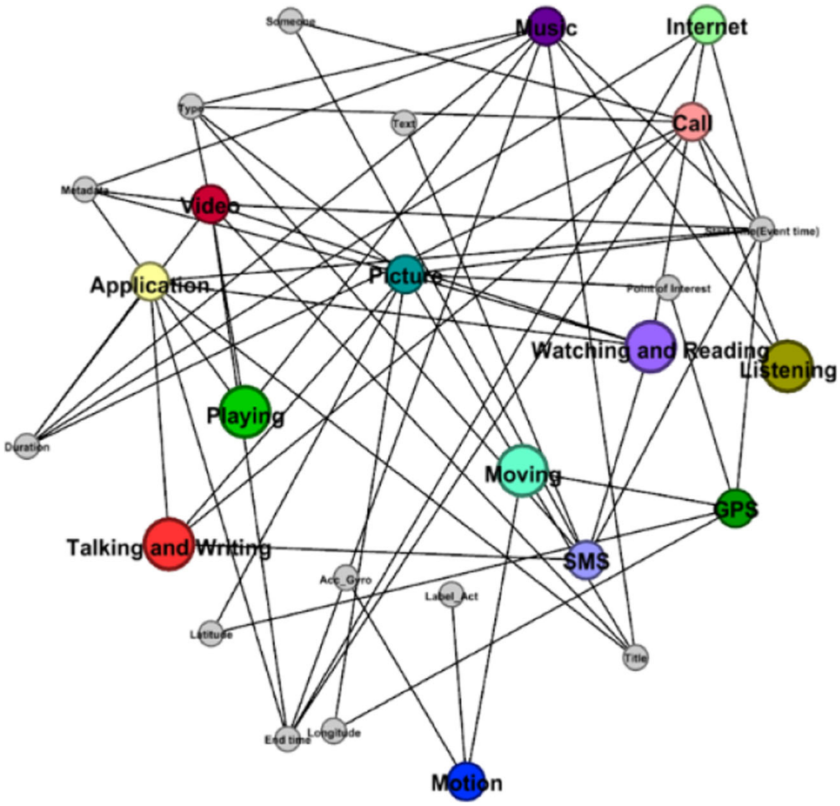
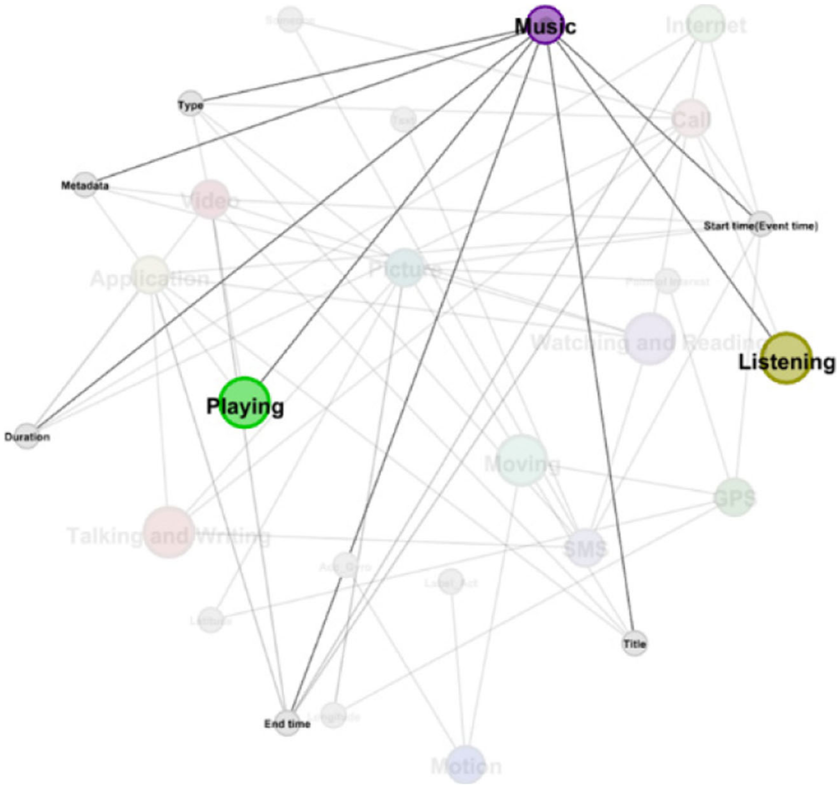


Fig. 14 Primary search process according to user's query



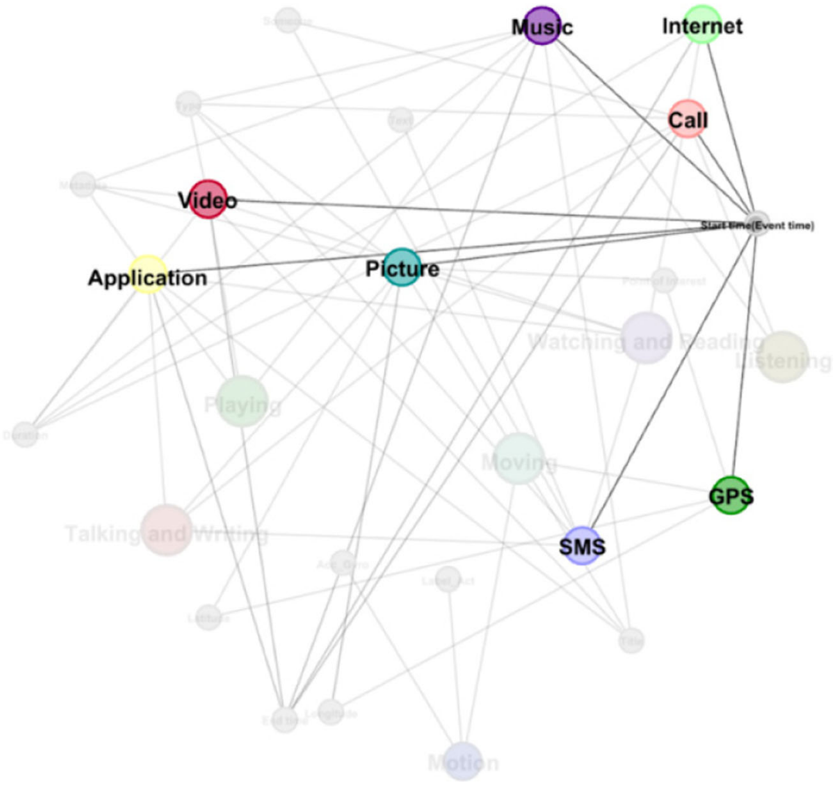


Fig. 15 Layer 2 category associated with start time node

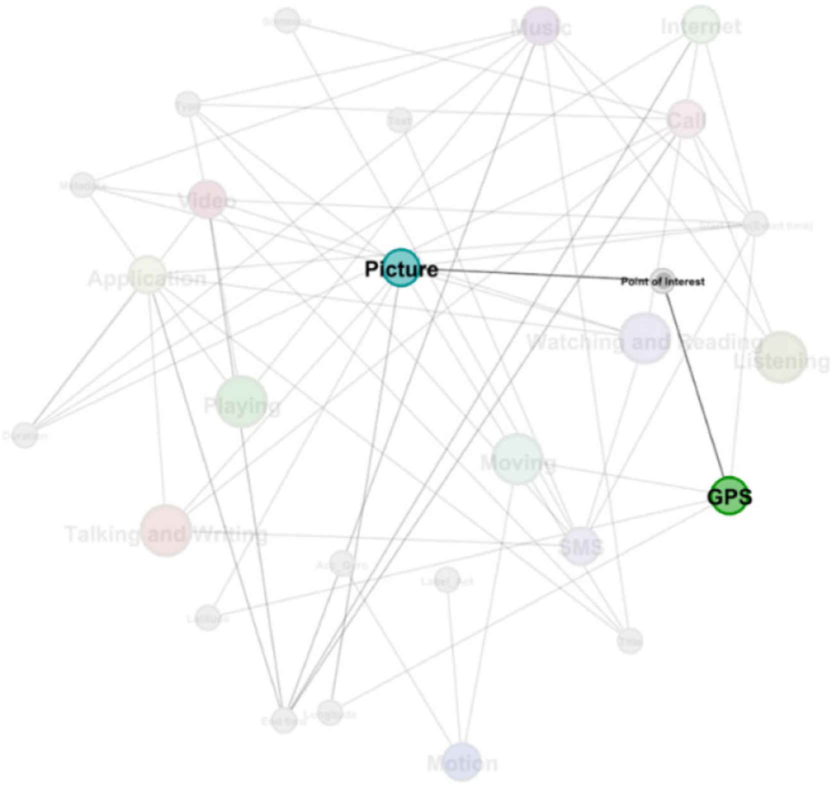


Fig. 16 Layer 2 category associated with location node

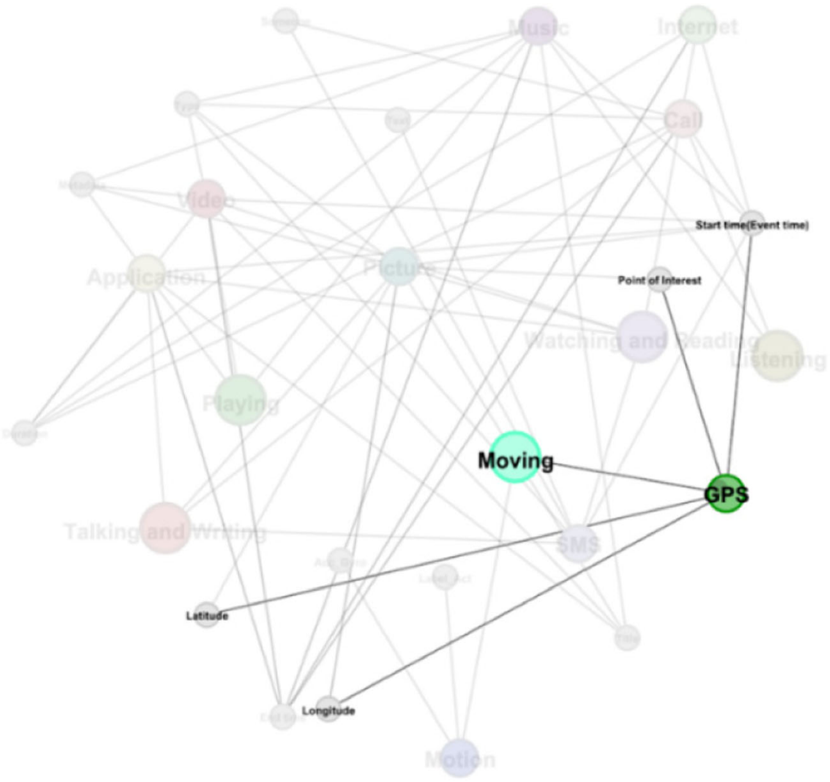


Fig. 17 Search for GPS-related nodes for results selection

Five users in each class generated training files and experimented. The figures in the table are the average values of the participants. With naive Bayes, the number of correctly classified instances are about 88% of the total. The reason for this is that the correct classification of walk is relatively low compared to other classes, because the values are not distinct compared to run or stop. Weight avg. represents the average value of the three classes and is compared with SVM based on this.

Experiments using the statistical-based SVM were also conducted. SVM is a classifier that classifies new input data by distance using the hyperplane found in the modeling process to find the optimal hyperplane to classify. Likewise, the algorithm is applied using the existing experimental structure and the seven attributes mentioned above.

Table 4 lists the results of applying the SVM to the experimental data. In the case of the SVM, there are parameters that can be adjusted. In this experiment, the adjusted values were the kernel and cost values. Gamma is a parameter to control the width of the Gaussian kernel and was fixed at one in the experiment. The cost (10, 20) plays a role in specifying the degree of overfitting, where larger values allow larger errors. In the case of the kernel (linear, polynomial, RBF (radial basis function)), there are many properties like the experimental data of this study, and it plays the role of mapping to a higher dimensional space when it is difficult to separate into one linear kernel. If all the data are mapped to a higher dimension, they can be divided into linear hyperplanes, which make it easier to separate all the data. Figure 18 shows the kernel's

Table 3 Naïve Bayesian experiment result

Correctly classified instances 88.2353%

Class	TP	FP	Precision	Recall	AUC
Walk	0.740	0.008	0.983	0.740	0.920
Run	0.987	0.171	0.771	0.987	0.931
Stop	0.942	0.007	0.980	0.942	0.988
Weight avg.	0.882	0.067	0.904	0.882	0.941

Table 4 Naïve Bayesian experiment result

Kernel	Gamma	Cost	Accuracy	AUC (weight avg.)
Linear	1.0	10	77.451	0.694
Linear	1.0	20	77.9412	0.694
Polynomial	1.0	10	83.8235	0.764
Polynomial	1.0	20	84.3137	0.882
RBF	1.0	10	86.2745	0.798
RBF	1.0	20	86.7647	0.901

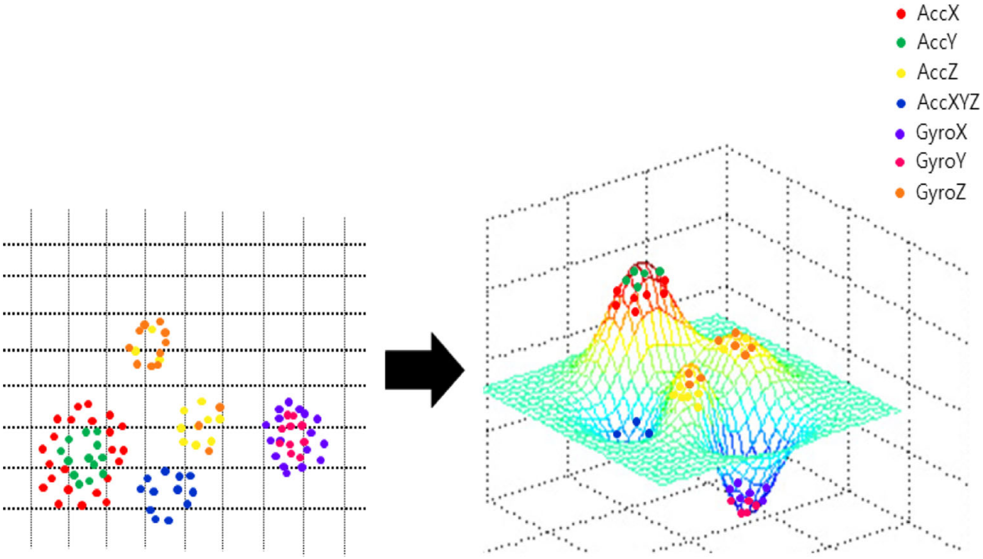


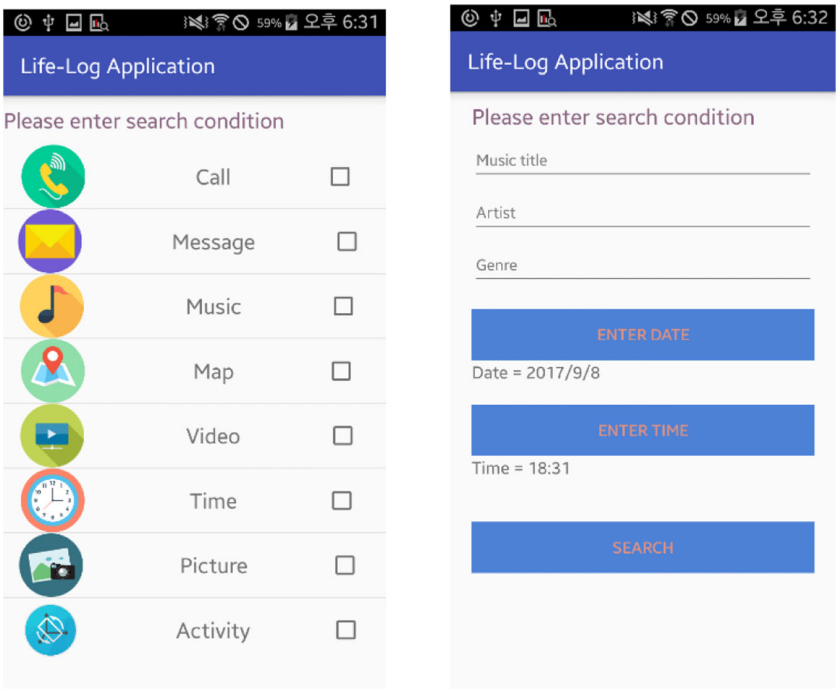
Fig. 18 Role of kernel

role, and Table 4 summarizes the results of applying the SVM to the experimental data in terms of the accuracy and AUC.

The experimental results showed that the SVM had the highest accuracy for the RBF kernel, but it was less than the naive Bayesian's accuracy. In the case of the AUC, the naive Bayesian showed the lowest value of 0.920 for walk, but it was higher than any experimental result of the SVM. Therefore, we found that naive Bayesian was efficient and effective for the intelligent behavior classification module. The algorithm was applied to the personal search system to classify the user's activity so that meaningful information could be stored.

4.2 Feedback and user satisfaction experiment based on semantic discovery

In order to effectively apply the lifelog management and retrieval system, feedback and user satisfaction tests were conducted through searches. We input a query from the user and checked the feedback time for the query and the user



59% 오후 6:31

Life-Log Application

Please enter search condition

Call □
Message □
Music □
Map □
Video □
Time □
Picture □
Activity □

59% 오후 6:32

Life-Log Application

Please enter search condition

Music title

Artist

Genre

ENTER DATE

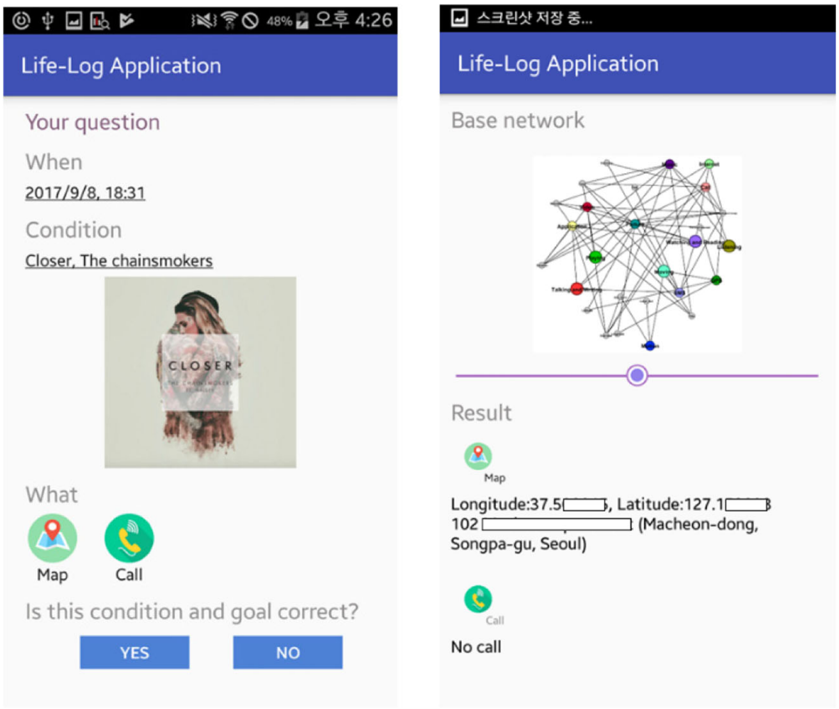
Date = 2017/9/8

ENTER TIME

Time = 18:31

SEARCH

Fig. 19 Question input screen



Life-Log Application

Your question

When

2017/9/8, 18:31

Condition

Closer, The chainsmokers

What

Map Call

Is this condition and goal correct?

YES NO

Life-Log Application

Base network

Result

Map

Longitude:37.5[redacted], Latitude:127.1[redacted]
102 [redacted] (Macheon-dong, Songpa-gu, Seoul)

Call

No call

Fig. 20 Query example screen (left) and search result screen (right)

satisfaction with the feedback result. Through this process, the reliability and accuracy of the system could be judged, and the reliability of the system could be verified.

To evaluate the user satisfaction, the user evaluated the usability and search-ability of the application and the system by filling out the questionnaire after using the lifelog management and retrieval system. Figure 19 shows an example screen for entering queries into the application and system. Figure 20 shows the result of the input and the result according to the input conditions. The user wondered about their position and telephone record when he/she listened to music. However, in the case of the telephone record, the result of "No call" can be seen because the user cannot listen to music and use the telephone at the same time.

The semantic discovery algorithm explained in Section 3.4 was internally implemented, and the interfaces are shown in Figs. 19 and 20. After each of the different input queries shown in Fig. 20 was performed three times,

Table 5 List of questions

Index	Question
1	When I heard "Closer" on "The chainsmoker" on September 8, 2017, what was my position and phone status?
2	What genre did I hear when you were active (walking, running) yesterday at 8 am?
3	Where do I usually use Internet applications?
4	What date and time did I call your parents (mother, father)?
5	Where is the point of interest on August 27, 2017?
6	What song did I hear when you passed around the point of interest on August 27, 2017?
7	What kind of logs were recorded at 9 am on May 23, 2017?
8	What time did I stay at "school" around 2/3/17 of 2017?
9	What are the most common actions (walking, running, and stopping) on August 30, 2017?
10	When was the last time a lot of photos were left?
11	Where is the point of interest on the day when a lot of photo records are left?
12	August 12, 2017 what kind of applications did I use the most?
13	When did I talk to the person you recently called?
14	What genres and singers do I listen to most?
15	What kinds of activities (walking, running, stopping) are mainly performed when listening to "ballads"?

Table 6 Average score by evaluation item

Index	Evaluation content	Average score
1	Is it convenient to check the records of each category?	7.40
2	Is it possible to intuitively perform the switching between screens?	8.00
3	Is the type of menu provided appropriate?	7.60
4	Is the application easy to use?	9.00
5	Are there enough functions for daily recording?	9.00
6	Is the collection appropriate during the recording process?	8.40
7	Is the content of the additional information extracted significant?	9.00
8	Is it likely to be useful for providing effective services using new information (behavior, points of interest) derived from the life log?	9.40
9	Is it easy to locate the menu?	7.60
10	Can you find the records you want easily?	8.00
11	Does the record check for each category work properly?	9.00
12	Does the association information obtained from the search through the network structure easily work for an individual?	9.00
13	Is the outcome of the search likely to be used effectively for future personalized service delivery?	8.60

the user filled out the user satisfaction questionnaire. Participants filled in their information by hand and responded to the inquiries and entered the evaluation items (Table 5).

For a detailed evaluation, the score was entered as an integer between 1 and 10, with a score closer to 10 indicating "very agree." Four usability items, five search-ability items—and the average score of each item was calculated. The results are listed in Table 6, and the average score was expressed by dividing the total score by the number of participants.

When the average score of the evaluation items in the user satisfaction questionnaire was expressed as a score of 10, the average of all the scores for 13 items was 8.462. The item with the lowest average score was "Convenient to check the records of each category?", and the highest item was "Possibility to be useful for providing effective service using new information (behavior, point of interest)?" The satisfaction with the ability to record or inquire about information and confirm new information was relatively high, whereas the scores on the visual screen delivered to the interface or user were relatively low compared to other items.

5 Conclusion

In this paper, we proposed a method to generate meaningful information by collecting, processing, and storing lifelog, which is the daily life data of smart phone users, and retrieving association information based on the network structure—classifies activities such as walking, running, and stopping using sensors related to movement.

In the proposed activity extraction module, we applied the naive Bayesian algorithm, which showed an excellent performance, with an accuracy of 88.23% and AUC value of 0.941. The POI extraction module implemented DBSCAN, a density-based clustering method that ignores noise and automatically generates clusters of various shapes according to the data, to represent individual characteristics. The two modules could provide meaningful information about the motion characteristics and the movement path of the user and can be used for more effective service than the lifelog using only the existing smart phone sensors.

Funding information This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2015R1D1A1A01059253).

References

1. The Nielsen Koreanclick (2017) 41st Population Estimation Survey Report. 2017.01. http://www.koreanclick.com/insights/newsletter_view.html?code=topic&id=433&page=1
2. Nejati H, Pomponiu V, Do TT ZY, Iravani S, Cheung NM (2016) Smartphone and mobile image processing for assisted living: health-monitoring apps powered by advanced mobile imaging algorithms. IEEE Signal Process Mag 33(4):30–48
3. Mafrur R, Nugraha IGD, Choi D (2015) Modeling and discovering human behavior from smartphone sensing life-log data for identification purpose. Hum Centric Comput Inform Sci 5(1):1–18
4. Ogata H, Hou B, Li M, Uosaki N, Mouri K, Liu S (2014) Ubiquitous learning project using life-logging technology in Japan. Educ Technol Soc J 17(2):85–100
5. Gemmell J, Bell G, Lueder R (2006) MyLifeBits: a personal database for everything. Commun ACM 49(1):88–95
6. Hodges S, Berry E, Wood K (2011) SenseCam: a wearable camera that stimulates and rehabilitates autobiographical memory. Memory 19(7):685–696

7. Doherty A, Caprani N, Óconaire C, Kalnikaite V, Gurrin C, Smeaton AF, O'Connor NE (2011) Passively recognizing human activities through lifelogging. Comput Hum Behav 27(5):1948–1958
8. Francese R, Risi M, Scanniello G, Tortora G (2016) Lifebook: a mobile personal information management system on the cloud. Proceedings of the International Working Conference on Advanced Visual Interfaces, pp 184–191
9. Tang D, Botzheim J, Kubota N (2015) Informationally structured pace for life log monitoring in elderly care. In: Systems, Man, and Cybernetics (SMC), 2015 I.E. International Conference on. IEEE, pp 1421–1426
10. Shah M, Mears B, Chakrabarti C, Spanias A (2012) Lifelogging: archival and retrieval of continuously recorded audio using wearable devices. IEEE International Conference on Emerging Signal Processing Applications, pp 99–102
11. Jalal A, Kamal S (2014) Real-time life logging via a depth silhouette-based human activity recognition system for smart home services. In: Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on, pp 74–80
12. Tsai MJ, Wu CL, Pradhan SK, Xie Y, Li TY, Fu LC, Zeng YC (2016) Context-aware activity prediction using human behavior pattern in real smart home environments. In: Automation Science and Engineering (CASE), 2016 I.E. International Conference on, pp 168–173
13. Hamm J, Stone B, Belkin M, Dennis S (2013) Automatic annotation of daily activity from smartphone-based multisensory streams. In: Uhler D, Mehta K, Wong JL (eds) MobiCASE 2012. LNICST 110:328–342
14. Semantic Web. https://www.w3.org/standards/semanticweb/
15. Maedche A, Maedche S (2001) Ontology learning for the Semantic Web. IEEE Intell Syst 16(2):72–79
16. Heflin J, Hendler JA, Luke S (2003) SHOE: a blueprint for the Semantic Web. MIT Press, Cambridge, pp 29–63
17. Hu C, Xu Z, Liu Y, Mei L, Chen L (2014) Xiangfeng Luo Semantic link network-based model for organizing multimedia big data. IEEE Trans Emerg Top Comput 2(3):376–387
18. Berners-Lee T, Hendler J, Lassila O (2001) The Semantic Web. Sci Am 284(5):28–37
19. Shadbolt N, Hall W, Berners-Lee T (2006) The Semantic Web revisited. IEEE Intell Syst 21(3):96–101