Next-Activity Set Prediction Based on Sequence Partitioning to Reduce Activity Pattern Complexity in the Multi-User Smart Space

Younggi KIM(a), Student Member and Younhee LEE(b), Nonmember

SUMMARY Human activity prediction has become a prerequisite for service recommendation and anomaly detection systems in a smart space including ambient assisted living (AAL) and activities of daily living (ADL). In this paper, we present a novel approach to predict the next-activity set in a multi-user smart space. Differing from the majority of the previous studies considering single-user activity patterns, our study considers multi-user activities that occur with a large variety of patterns. Its complexity increases exponentially according to the number of users. In the multi-user smart space, there can be inevitably multiple next-activity candidates after multi-user activities occur. To solve the next-activity problem in a multi-user situation, we propose activity set prediction rather than one activity prediction. We also propose activity sequence partitioning to reduce the complexity of the multi-user activity pattern. This divides an activity sequence into start, ongoing, and finish zones based on the features in the tendency of activity occurrences. The majority of the activities in a multi-user environment occur at the beginning or end, rather than the middle, of an activity sequence. Furthermore, the types of activities typically occurring in a multi-user environment can be sufficiently distinguishable. Exploiting these characteristics, we suggest a two-step procedure to predict the next-activity set utilizing a long short-term memory (LSTM) model. The first step identifies the zones to which current activities belong. In the next step, we construct three different LSTM models to predict the next-activity set in each zone. To evaluate the proposed approach, we experimented using a real dataset generated from our campus testbed. Our experiments confirmed the complexity reduction and high accuracy in the next-activity set prediction. Thus, it can be effectively utilized for various applications with context-awareness in a multi-user smart space.

key words: activity prediction, sequence partitioning, multi-user smart space, LSTM

1. Introduction

As the Internet of Things (IoT) continues to improve the quality of human life for a new generation, the interest in context-aware technology has grown over the last decade. In this IoT era, many studies have focused on achieving context-awareness in a multi-user smart space. Di          

To perform activity prediction, many researchers have proposed methods based on machine learning algorithms. In [22] they addressed a prediction algorithm for smart home inhabitants based on a Markov model. Another study [26] claimed to use the current activity, including location and time features, to solve daily activity prediction problems based on a Bayesian network. Many research studies have focused on activity prediction using different methodologies including regression [25] and decision trees [17]. However, the majority of the existing works focuses on single-user activity patterns in the fields of ambient assisted living (AAL) and smart home resident activities of daily life (ADL). In these situations, activity sequences are relatively simple and repetitive. Nevertheless, their accuracies demonstrated a limitation in the inference.

In the multi-user smart spaces, the characteristics of activity occurrence are considerably different from the single-user environment. In this environment, activity occurrences involve a common intention, e.g., a meeting, a presentation, or having a meal. We define an intention as a task that is composed of a sequence of activities generated by a group of people. For example, activities such as entrance, turning on a projector, and sitting down can occur for a task such as a meeting. When a task occurs, the group of people often acts simultaneously. So, the activity patterns in a smart space become more diverse as the number of people and the types of activities in a task increase. Theoretically, patterns in a smart space can have the complexity $A_{all}^N$, where $A_{all}$ represents the number of all types of possible activities in a location, and $N$ is the number of people. Therefore, it is more difficult to predict the next activity as the number of people and activity types increase. The multiple series of activities cause different next activities after a sequence of activities. It is for this reason that activity set prediction is necessary in a multi-user environment, not a single activity prediction.

If recognizing the task of activity occurrences is possible in an early stage, the number of possible activities can be reduced. However, this is an extremely difficult problem with only limited information in the early stage. That is, since multiple next activities are possible in the early stage, it requires rather considerable amount of time to figure out the contour of a task and until then, next-activity related services cannot be recommended. That is why we focused on activity set prediction rather than a task recognition.

Instead of recognizing the task, we propose activity sequence partitioning, which divides a sequence into start,
ongoing, and finish zones to reduce the pattern complexity. When multiple users generate activities having the same intention, the majority of the activities typically occur in either the beginning or end, whereas activities occur relatively sparsely in the middle. Certain types of activities generally occur at the beginning, while others typically occur in the middle or end part of a task. Thus, the types of activity are generally correlated with the zone, so they are reasonably distinguishable according to the three zones. Based on these features, an appropriate zone division can reduce the possible activity types ($A_{zone}^N < A_{all}^N$, where $A_{zone}$ is the number of the types of possible activities in a certain zone).

Exploiting the activity sequence partitioning to the fullest, we propose a two-step procedure for prediction: 1) the zone recognition step and 2) the next-activity set prediction step. Although the pattern complexity in each zone can be reduced by the partitioning, the next predicted activity cannot be confined to a single activity; several activities can meet to a contextual situation. Thus, the model needs to predict the possible activity set instead of a single activity in the second step.

To recognize the current zone in a task, a portion of an activity sequence up to the window size is used as input. Then, the zone recognition model outputs the zone sequence corresponding to the input activities. When it classifies the current zone, it passes the portion of activities that belongs only to the current zone among all of the input activities to the corresponding long short-term memory (LSTM) model for the next activity prediction. As the LSTM model for each zone learns the next activity sets, it predicts next activity candidates when it receives the input activities passed from the first step.

We evaluated the proposed approach using real data generated from the testbed, which collected multi-user activities for eight months. Using an entropy evaluation, it was confirmed that the complexity of the sequence partitioning could be reduced by approximately 10% compared to the case without partitioning. Moreover, the types of possible activities in each zone were reduced when the sequences were divided into the three zones. Consequently, the accuracy of the next-activity set prediction with the sequence partitioning was greater than that without sequence partitioning. The results confirmed that the next-activity set could be predicted with sufficiently high accuracy for use in applications with context-awareness, even when the environment was composed of multi-user activity sequences and different intentions. Based on a next-activity set prediction, useful applications can be realized. For example, an actuator service related to the next activity can be executed autonomously or recommended to users. Also, for AAL, anomaly activities can be analyzed by checking whether the next activity is out of the predicted activity set.

The remainder of this paper is organized as follows: Section 2 discusses activity prediction related works and differences with the proposed approach; Section 3 analyzes the problem we address; in Sect. 4, we describe the two-step procedure for next activity set prediction in a multi-user smart space; in Sect. 5, we evaluate our approach using our testbed; and finally, we conclude in Sect. 6.

2. Related Works

Over the last decade, the field of activity recognition has received significant attention [12], [16], [18], [20], [32], [34]. For further studies, the activity prediction field has attracted serious consideration and has been approached with various technologies, including sensors, wearable devices, mobile devices, images and video [5], [19], [21], [28]. In [10] and [13], they focused on recognizing group activities from wearable sensors, which was similar subject with this paper. The results showed high accuracy to recognize the group activities. From our point of view, they recognized the intention of multi-user behaviors captured by each wearable devices. In our paper, we focused to predict the next behaviors from installed IoT devices without the intention information.

Many researchers have proposed their methodologies to solve the emerging problem of activity prediction. The majority have focused on the single-user environment, particularly AAL and ADL routines. The research in [27] addressed a Markov model-based method derived from prediction by partial matching (PPM) that selects the most probable future activity. Because of the limitations based on simple sensor data, this method was modified to improve the accuracy by adding a time component [22]; however, the resulting study was limited because it considered only one specific situation: one person in one room. Other machine learning algorithms have been presented for activity prediction. [26] constructed a Bayesian network and utilized three features – activity location feature, activity time of day feature and activity day of week feature – with the current activity used to predict the next activity. They focused on single-resident smart home apartment environments and conjugated daily/weekly routine information. In [24], the authors proposed an activity forecasting method using a regression tree. They considered elderly adult ADL, which is similar to the dataset used in [26], such that a single-user activity routine was the major target for the prediction. Another regression method was addressed in [25], which dealt with smart home activity prediction from real sensor data. Even though they considered multiple participants, the majority of the activities had a high dependency on location, i.e., location-occupied data could have a strong influence on the next activity prediction. A small number of studies have addressed the multi-user environment and various intentions.

Recently, dramatic developments in machine learning have been made in diverse areas. Specifically, a recurrent neural network (RNN) model has been proposed to address time-series data. Many recognition fields such as speech, language, and image have utilized RNN models [1], [7], [11], [23], [29]. However, RNN models demonstrate the weakness of long length of sequence inference, the so-called vanishing gradient problem. To resolve this prob-
lem, LSTM models have been suggested as an alternative. By imposing the forget gate concept, LSTM overcomes the vanishing gradient problem [14]. LSTM performs more effectively and thus, is more widely utilized for inference in speech [33], image [6], and text labeling [2].

The LSTM model can be designed in different forms according to its purpose. As Fig. 1 depicts, the LSTM model is composed of three layers: input, hidden, and output layers. In general, the dataset is divided into training data and testing data. Both data are composed of input and output pairwise. With the training data, an LSTM model learns the output data as answers, corresponding to the input data. As it learns the training data, the parameters in the hidden layer are fitted to the data. After the learning, the accuracy can be evaluated using testing data. The model can be designed in a many-to-many approach, where the output has the same length as the input like language translation, or a many-to-one approach for the purposes of recognition or classification. In this paper, we adopt and utilize an LSTM model for time-series data prediction. We use both approaches of LSTM to construct the next activity prediction method in a two-step process.

3. Attributes of Multi-User Smart Space

Research attention is typically directed towards the following three types of smart spaces: smart homes, smart offices, and smart cities. Complexity and diversity are different in each environment. In smart spaces with a single-user, the activity patterns usually have relatively simple and repetitive daily or weekly routines. Conversely, more complex and diverse activity patterns occur when the space is occupied by multiple users with different intentions. In multi-user environments, multiple activities are generated simultaneously in any sequence. We define a series of activities having the same intention as a task, for example, a meeting, presentation, or having a meal. A combination of activities such as entering, sitting down, turning on a projector, and exiting compose a task.

- **Task**: A series of activities having the same intention.
- **Activity**: The behavior of a person.

Within diverse activity patterns, we found out the attributes of activity sequence patterns of a multi-user space as follows. These attributes never appear in a single-user smart space:

- **Multiple next activity candidates attribute**. In a multi-user smart space with a specific intention, multiple activity candidates can occur after an activity sequence. For example, after the activity sequence \{open a door, entrance, sit down, entrance\}, the next activity can be among various activities such as entrance, close a door, turn on a projector, etc. due to multi-user activities.
- **Sub-sequence attribute**. A relatively short-length activity sequence of a task could occur as a sub-sequence in other longer tasks, making it difficult to recognize whether a task is finished or ongoing. For example, the activity sequence in a task, phone call, can be \{entrance, sit down, stand up, exit\}. However, the same activity sequence can occur at the beginning of a task such as a meeting.
- **Multi-semantics attribute**. The same activity sequence could have different meanings according to the zone. For example, a sequence of activities such as standing up and exiting, in the middle of a task could indicate going to restroom for a while, whereas the same sequence at the end of a task could mean finishing the task.
- **Interleaved sequence attribute**. In a public space, it is possible for more than two tasks to exist at the same time and location. Thus, more than two activity sequences for different intentions can co-exist in a location.

In this paper, we consider the above attributes except for the interleaved sequence attribute. In a public smart space, there exist concurrent tasks with a considerably greater number of people and the environment includes all of the above attributes. This public smart space environment is beyond the scope of this paper. However, it could be considered in a future work.

3.1 Next-Activity Set Prediction in Multi-User Smart Spaces

The next activity candidates attribute is one of the most differing attributes from a single-user environment. There are several possible activities that can occur after an activity sequence as multiple users generate activities simultaneously. Therefore, the next-activity set prediction method that we propose is indispensable in such an environment. For the elaborate description, the next-activity set is defined as all possible activities after an activity sequence. The next activity candidates are the output of the prediction.
3.2 Pattern Complexity Analysis for the Activity Sequence Partitioning

Theoretically, the complexity of the activity sequence patterns is proportional to the number of people and the types of activities \( A_{all} \), where \( A_{all} \) represents the number of all types of possible activities in a location, and \( N \) is the number of people. If it is possible to recognize the intention of activities, the complexity can be reduced. However, it is extremely difficult to recognize the intention from the partial information available at the early stage.

Instead of the intention recognition approach, we designed an effective approach using the properties regarding activity occurrences. Activity occurrences have prominent properties in terms of the frequency of activities and types of activities. The majority of activities typically occur at the beginning or end of a sequence. In the start zone, people gather and prepare to start a task. Similarly, people start to break up a task in the finish zone. Conversely, relatively lesser activities occur in the middle of a task. Based on this trend, we propose the partitioning of the activity sequences to reduce the pattern complexity, referred as sequence partitioning in short. To separate each sequence, the boundaries between each part must be determined. However, setting the criteria for the boundary is difficult and ambiguous, even if a person observes the entire activity sequence for a division.

Alternatively, we divide each activity sequence in even time length into 3 to 15 parts and determine the trend of the frequency of the activity occurrences. From the dataset, all the activity sequences are divided evenly, and the average numbers of activity occurrences are calculated as illustrated in Fig. 2 with 4 cases. In the Figure, each divided part in a division case represents the number of activity occurrences averaged from the corresponding portion of all the activity sequences. That is, each part contains the frequency information at a specific range in the progression of activity sequences. As expected, the average numbers of the first and last parts were the highest in all division cases as indicated in Fig. 2. Thus, the tendency of a high frequency of activity occurrences at the first and last parts was confirmed. Through sequence partitioning, the complexity can be significantly reduced, considerably less than that without partitioning: \( A_{part} < A_{all} \), where \( A_{part} \) is the number of types of all possible activities in a certain part. Based on sequence partitioning, we propose a two-step procedure for the next activity set prediction.

4. Next Activity Set Prediction Scheme

Based on the sequence partitioning detailed above, we present a two-step procedure consisting of zone recognition and next activity set prediction, as illustrated in Fig. 3. The zone is defined as a specific part or a set of parts in a division of the activity sequences, so it implicitly occupies a specific time portion within the activity sequences. As the first step, the current zone should be recognized; then, the next-activity set can be predicted according to the recognized the current zone.

To determine the appropriate divisions with the activity sequences, the tendency that the first and last parts have the highest average number of activity occurrences should be considered. The division of activity sequences becomes more distinguishable when the difference of mean frequencies between adjacent parts becomes larger. A variance of each part is also a useful criterion for a division because the smaller variance of a part signifies the lesser difference between activity frequencies in that part. To represent these attributes, a Z-test [30] is utilized as follows:

\[
Z = \frac{M_i - M_{i+1}}{\sqrt{\frac{\sigma_i^2}{N_i} + \frac{\sigma_{i+1}^2}{N_{i+1}}}}, \quad i = i - \text{th part}
\]

where \( M_i \) and \( \sigma_i \) are the mean value and the standard deviation respectively of the activity occurrences in the \( k \)-th part of all activity sequences. And \( N_i \) is the number of activity occurrences in the \( k \)-th part. If \( Z \)-value is large, the numbers of activity occurrences in neighboring parts are relatively uniform and the differences between the neighboring parts are more distinguishable. Thus, it is appropriate to determine a boundary when a \( Z \)-value is larger. To apply the Z-test, the two largest \( Z \)-values must be considered to re-

![Fig. 2](image1) Average frequency of activity occurrences in each division

![Fig. 3](image2) Overview of two-step procedure
reflect the frequency tendency which the first and last parts are outstanding values as in Sect. 3. We utilize the sum of the largest two Z-values in determining the most appropriate number of divisions. In the divisions, the last Z-value is a negative number; therefore, we use its absolute value. Figure 4 depicts the sum of two largest Z-values.

According to the Z-test result, the sum of the two largest Z-values is maximized when a task is divided into eight parts with equal time periods in various division cases. Thus, dividing by eight is the most appropriate to determine the boundaries of zones in our dataset. The appropriate division number can be varied according to the Z-values of the dataset from other environments. Since the computed Z-values except the first and last are uniformly small, we aggregate the second through seventh zones to comprise the ongoing zone. The first and last parts are designated as start and last zone respectively so that there are only three parts—start, ongoing, and finish zones—in each task.

4.1 Zone Recognition Step

Based on sequence partitioning, an LSTM model is designed to learn the correct zone sequence using a many-to-many approach when it receives a sequence of activities as an input. Then, the output is a sequence of the zones that has the same length as the input activities. For example, if {Entering, Turning on a light, Sitting down, Turning on a projector} activities occur, the result of this step can be [start zone, start zone, start zone, ongoing zone]. By accumulating the outputs of the zone sequence, the current zone is determined to be ongoing zone. Our implementation passes the activities belonging to the current zone to the second step procedure for the activity prediction.

There are two possible approaches by which the model can learn the time-series data: time window-based learning and incremental window-based learning. The time window-based learning approach receives a sequence of activities as input and outputs a zone sequence corresponding to the time window size. The incremental window-based learning extends the window size from three to the end of the activity sequence, increasing by one at each input. In the LSTM model, the input length determines the model size. The longest length in the time window-based learning approach is always equal to the time window size. Conversely, the length in the incremental window-based learning is considerably longer because the LSTM size must be at least as long as the longest sequence of data. Because the latter approach has more data to infer from an input, the accuracy is higher than the former approach; however, the learning time is considerably longer. Because the real-time activity prediction performance is not related to the learning time, incremental window-based learning is a superior approach for improved performance.

4.2 Next-Activity Set Prediction Step

According to the division of the zones, three different LSTM models for the zones are constructed. When the current zone is recognized in the first step, a chunk of the activity sequence is passed to the second step. A chunk of the activity sequence is a portion of one entire sequence of activities. In the previous step, the current zone is already recognized by the LSTM having the input of occurred activity sequence. Among the occurred activities, those activities that belong to the current zone are chunk to be passed to the second step. In this step, LSTM models use the many-to-one approach to predict the next-activity set. If a task is not separated into zones, the next candidate activities vary widely because of the characteristics of the multi-user activity occurrences. Conversely, the number of candidate activities with sequence partitioning can be reduced because the next candidate activities are reasonably specialized to each zone.

However, even if an activity sequence is divided into three zones, there can be still multiple candidate activities in each zone. Therefore, each model is forced to learn several candidate activities based on their occurrence probabilities in the training data. For example, if there are four activity sequences in the training data, \{activity1, activity2, activity3, activity1\}, \{activity1, activity2, activity3, activity1\}, \{activity1, activity2, activity3, activity2\}, and \{activity1, activity2, activity3, activity3\}, the next activity of the sequence \{activity1, activity2, activity3\} is among activity1, activity2, and activity3 having different ratios of occurrences. We used this ratio as output instead of one-hot vectors, while the input data are one-hot vectors. If there are 5 types of activities, the one-hot vector of activity1 can be expressed as \[1, 0, 0, 0, 0\]. Instead of the one-hot vector, we set the value of occurrence ratio as output, for example, the next activity set of the sequence \{activity1, activity2, activity3\} is expressed as \[0.5, 0.25, 0.25, 0, 0\].

Because the frequency of activity occurrences is high in the start and finish zones, the intervals between their activity occurrences are relatively short. Conversely, they are relatively long in the ongoing zone. Thus, the different time window sizes are suitable for high accuracy in each LSTM model. The time window size for each zone was empirically determined; the results of the evaluation are described in the next section.

5. Experiments and Evaluation

To validate the proposed model, LSTM-based two-step
models were developed using TensorFlow [31], one of the most widely used open source libraries for machine learning. As mentioned above, the zone recognition procedure was developed using the many-to-many approach of the LSTM model. The next-activity set prediction procedure was designed using the many-to-one model. Further, the second step included the functionality to select the Top-K candidate activities in the order of highest probabilities. To apply the two-step process, we used our testbed and built a dataset of real data. In this section, the experimental setup and evaluation results are described.

5.1 Experiment Setup

To test the proposed approach, multi-user activity data were collected from our testbed. We constructed a testbed of a smart meeting room on the campus with various IoT devices installed. The devices included in the testbed were four actuators and eight sensors, as illustrated in Fig. 5. When people performed activities in the testbed, their occurrences were sensed by the devices and stored as activities in a MongoDB database [4]. All the actuator devices were controlled by Raspberry Pi boards. The sensors generated the sensed data every three seconds. A webcam was used to confirm ground truth and revise the data. Some activities such as clean a desk, raise a curtain, lower a curtain, open a window, and close a window could not be captured by sensors. So we inserted those activities by verifying the webcam records, and those activities constituted about 0.5% of total activity occurrences. The records presented in Table 1 contain 414 task occurrences from September 2015 to June 2016. Ten types of tasks were obtained from a combination of 23 types of activities. The longest task was a study that took approximately eight hours, although it was composed of only 24 activities. The maximum number of people was 14 at a lab seminar.

Each sequence of activities for a task was segmented based on the presence of people. When a person entered the empty room, a task started and a task was finished when there was no person. However, even if there was no person, the task may no be finished due to a break time, etc. So, through checking webcam records, we concatenated activity sequences when the task actually continued. And we deleted an activity sequence when there was no task, for example, a person entered and exited without any intentions. We also eliminated all interrupted data, e.g., when people were about to start a certain task and another group entered or when a task stopped because of a room reservation by someone else. We also excluded concurrent tasks data, which is beyond the scope of this paper.

In each task, the data included not only the activity sequence but also additional information such as time and...
number of people. Even though we proposed a two-step prediction model, we used those additional information in the first step only, i.e. current zone recognition. We experimented whether the additional information influenced on the performance. Thus, an activity sequence with additional data was considered in the current zone recognition. The input data was the concatenation of the activity’s one-hot vector and additional data. For example, the activity entrance can be expressed as \([0, 0, 1, 0]\) by one-hot vector. When additional data such as time and the number of people are attached for input data, the vector is transformed into \([0, 0, 1, 0, 0.58, 3]\) (0.58 means the time 14:00, derived by \(14\times 60\) min/24\(\times 60\) min, and 3 is the number of people). We divided the dataset into 90% training data and 10% testing data.

5.2 Complexity Reduction by Sequence Partitioning

The next-activity set prediction is based on the sequence partitioning. We expected that the partitioning would make the zone specific attributes distinguishable. As expected, the frequency of each zone had a specific trend. We divided each sequence into eight parts evenly which is the optimized division; the first and last parts were labeled as start zone and finish zone, respectively, and the remainder of the divisions were aggregated to be ongoing zone.

The most frequent activities were Enter and Sit down in the start zone. The probabilities of the two activity occurrences were 33.21% and 24.52%, respectively in that zone. In the finish zone, Exit and Stand up were the most frequent activities and their occurrence probabilities were 33.20% and 28.19%, respectively. Sit down, Enter, and Stand up activities were most frequent in the ongoing zone. Their probabilities were 23.61%, 21.02%, and 18.01%, respectively. Thus, the most frequent types of activities were distinguishable in each zone.

In Sect. 3, we mentioned that the complexity of the sequence patterns is \(A_{all}^{N}\), where \(A_{all}\) is the number of all types of possible activities and \(N\) is the number of people. All possible activity types were 23 \((A_{all} = 23)\) in our testbed. The possible activities in start zone and finish zone were 19 \((A_{start\ zone} = A_{finish\ zone} = 19)\). There were 21 possible activities in the ongoing zone. We calculated the entropy of activity occurrence.

\[
\text{entropy} = -\sum_{i} p(a_i) \cdot \log_2 p(a_i)
\]

where \(p(a_i)\) is the probability of activity \(a\) occurrence. The entropy was 2.79 in the start zone and 2.66 in the finish zone. The entropy in the ongoing zone was 2.98. It was 3.11 without the partitioning; thus, the sequence partitioning achieved lesser entropy by approximately 10%. The reduced entropy, therefore, proved that the complexity was reduced by the sequence partitioning that we propose herein.

5.3 Zone Recognition Step

As explained in Sect.4, there are two approaches to implement the LSTM model for learning the zone recognition: time window-based learning and incremental window-based learning. We have conducted these approaches and compared their F-measures. We conducted an incremental window-based learning model that had an activity sequence as its input. That is, its input sequence was increased by one activity at each training until a task was finished.

To evaluate the zone recognition performance using the testing data, F-measure was adopted as the performance metric. The F-measure of the incremental window-based learning was as high as 79.28%. The time window-based learning had input activities that were accumulated as large as the time window size. We experimented with every window size from 20 to 300 s in increments of 10 s. The F-measure was about 70.69%, as indicated in Table 2. Because the incremental window-based learning refers more previous activity occurrences, the F-measure of it was higher than the time window-based learning. When the zone was recognized by the time window-based learning, its input activities were the partial activity sequence of a task. Because the partial sequence could occur in another zone of a task, it would be difficult for the time window-based learning to recognize the zone accurately.

We additionally conducted and compared these results with those cases that appended additional information in both models. The additional information included the frequency of the activity occurrence, the interval time between two adjacent activities and the number of people. However, these additional information did not improve the performance. Thus, these additional information may not have close relationship with a specific zone.

5.4 Next-Activity Set Prediction Step

We designed the LSTM model to predict the Top-K candidate activities, which means the \(K\) number of next activities with the highest probability. When the true answer was contained in the Top-K candidates, we determined the prediction to be correct. This is defined as Hit@\(K\). We evaluated the accuracies of Hit@5 and Hit@3 for each zone model.

First, we constructed an LSTM model that predicted only one next activity. Then, we constructed an LSTM model that predicted an activity set (Top-5 and Top-3) without sequence partitioning and compared it to the model that predicted Top-K candidates with sequence partitioning. In predicting one activity, there was low accuracy predicting the next activity, as we expected. The accuracy was approximately 50%. This result signified that predicting one next activity is challenging problem due to the diverse activity patterns by the multiple users. In predicting activity set, the
patterns without a sequence partitioning were still diverse that the highest accuracies of the Top-5 and Top-3 were at most 82.39% and 78.25%, respectively. As mentioned, the higher pattern complexity of the baseline model made it difficult to predict accurately, even if it predicted several next activities.

The zone-based model with sequence partitioning approach demonstrated higher prediction accuracies. Each zone model in this step performed learning with data that were assumed to be contained in each correct zone. In the start zone, the highest accuracy was 94.54% in case of the Top-5 when the window size was 70 s. When it predicted the Top-3 candidates, the accuracy was as high as 92.15% when the window size was 80 s. In the ongoing zone, the accuracy was as high as 92.86% with the Top-5 and 90.47% with the Top-3 when the window size was 130 s. In the ongoing zone, fewer activities occurred and therefore the interval time between adjacent activity occurrences was long. In the finish zone, the highest accuracies were 95.58% and 89.53% when the model predicted the Top-5 and Top-3, respectively. Similar to the start zone, a large number of activities occurred almost simultaneously in the finish zone. The results indicated that sequence partitioning contributes to improved approximately 13% prediction accuracy.

In the three zones, the window sizes with the highest accuracies were similar to their average time intervals respectively. This can be interpreted that if the window size is larger than the average time interval, it has lesser the volume of test data and lesser learning opportunities for encountering the variety of next activity. If the window size is smaller than the average time interval, the window size can break up a series of activities within the average time interval which may have close relationship between themselves, so that the learning performance become ineffective.

Our evaluations were conducted performing the two step independently in the above procedures. We also evaluated the accuracy of the two-step process as a whole, using the optimized results obtained from each step. In detail, based on the incremental window-based learning in the first step, we chose a 70 s window size for the start and finish zones and a 130 s window size for the ongoing zone in the second step. The resulting accuracy was at most 83.16% slightly above the zone recognition accuracy, which means that the second step outputs correct answers on some of the zones that the first step incorrectly determined. This implies that if the first step is improved, then the final accuracy will be certainly improved, so our future work will focus on the refining the first step.

Despite of that, these results confirmed that the proposed method achieved the performance comparable with the previous studies focusing on a single-user environment and can be applied to the services of a multi-user environment such as a service recommendation system.

6. Conclusion and Future Work

We considered an environment with multiple users and various intentions. A task is composed of a series of people activities having the same intention. In this situation, the activity patterns with multiple users were overly diverse to enable prediction of the next activity. Although the LSTM model is widely used to predict time series data, environmental attributes should be considered when it is applied. We considered the tendencies in the frequency of multi-user activity occurrences and proposed the sequence partitioning based activity prediction. This sequence partitioning led to each task being properly divided into three parts: start, ongoing, and finish zones.

We designed a two-step process to recognize the current zone and to predict the next candidate activities. To
evaluate the proposed approach, we developed a real dataset generated from our testbed. In the current zone recognition step, we experimented using incremental window-based learning and time window-based learning. The results indicated that incremental window-based learning recognized the current zone more accurately because it refers more input activities than time window-based learning.

In the next activity candidates prediction step, we evaluated the accuracy using the concept of Hit@K, assuming the current zone is correctly recognized. We designed the prediction model for each zone learning from the zone specific data to predict the Top-K candidates. When the current zone was assumed to be correctly recognized, the accuracy in each zone demonstrated the highest performance when the time window size in each zone was similar to the average interval time of that zone. When we used the current zone determined from the first step, the final accuracy was 83.16%. In conclusion, the division of zones with a two-step process achieved a high accuracy in a multi-user smart space, comparable to studies in a single-user environment.

If a zone is recognized more accurately with other context-related information, the activity set prediction can be strengthened. We have been accumulating the real data. A greater amount of data will improve the performance. Contextual information and more data captured automatically by IoT devices will be considered in our future works. Furthermore, several tasks can concurrently exist at the same location. In this situation, the task segmentation method is necessary to classify groups of activities, which is our future work. We have studied the task segmentation for non-realtime data [3]. If distinguishing the concurrent tasks by some classification method is realized in real-time, the next activity can be predicted even if the smart space is occupied by multiple users performing multiple tasks simultaneously.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2016R1A2B4014688).

References


Younggi Kim is a Ph.D. student at the School of Computing at KAIST, South Korea. He received B.E. from Korea University, South Korea in 2010 and M.S degree in Computer Science from KAIST, South Korea in 2012. His research interests include machine learning, Internet of things, and human activity.

Younghee Lee received his B.S degree in Electronic Engineering from Seoul National University in 1976, and M.S degree in Electronic Engineering in 1980. Then, he received Ph.D degree in Computer Science from France UTC (Universite de Technologie de Compiqne) in 1984. He had worked for ETRI from 1984 until 1997, and then had been the professor in ICU from 1998 to 2010 until merger with KAIST. He is now the professor of the Computer Science department in KAIST, and his interested fields are the Computer Networks, Overlay Networks, Mobile Computing, Pervasive Computing and Future Internet.