The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Generating an Event Timeline about Daily Activities from a Semantic Concept Stream

Taiki Miyanishi,[1] **Jun-ichiro Hirayama,**[2,1] **Takuya Maekawa,**[3,1] **Motoaki Kawanabe**[1,2]
[1]Advanced Telecommunications Research Institute International (ATR), Kyoto, Japan
[2]RIKEN Center for Advanced Intelligence Project (AIP), Tokyo, Japan
[3]Graduate School of Information Science and Technology, Osaka University, Osaka, Japan
{miyanishi, hirayama, t.maekawa, kawanabe}@atr.jp

Abstract

Recognizing activities of daily living (ADLs) in the real world is an important task for understanding everyday human life. However, even though our life events consist of chronological ADLs with the corresponding places and objects (e.g., drinking coffee in the living room after making coffee in the kitchen and walking to the living room), most existing works focus on predicting individual activity labels from sensor data. In this paper, we introduce a novel framework that produces an event timeline of ADLs in a home environment. The proposed method combines semantic concepts such as action, object, and place detected by sensors for generating stereotypical event sequences with the following three real-world properties. First, we use temporal interactions among concepts to remove objects and places unrelated to each action. Second, we use commonsense knowledge mined from a language resource to find a possible combination of concepts in the real world. Third, we use temporal variations of events to filter repetitive events, since our daily life changes over time. We use cross-place validation to evaluate our proposed method on a daily-activities dataset with manually labeled event descriptions. The empirical evaluation demonstrates that our method using real-world properties improves the performance of generating an event timeline over diverse environments.

Introduction

Recognizing human activities and his surrounding situation in the real world is an important task for understanding everyday human life. The purpose of such activity recognition is to automatically discover and identify human activities through mining signals from a wide variety of pervasive and wearable sensors. With the advances in accessible sensor technology, many real-world applications of activity recognition have been proposed such as monitoring Alzheimer's disease patients (Meditskos, Kontopoulos, and Kompatsiaris 2014), discovering activity patterns in a smart home (Rashidi and Cook 2010), recognizing nursing activities for improving medical care (Inoue et al. 2015), and carrying out life-logging functions (Castro et al. 2015).

Currently, many works are tackling these activity recognition problems (Bao and Intille 2004; Ramanan 2012; Patterson et al. 2005; Maekawa et al. 2010; Ordóñez and Roggen 2016) by classifying sequences of sensor data into discrete labels identifying activities of daily living (ADLs). These are basic activities in human living such as "walking," "eating," and "reading a book." However, simply predicting ADL labels is not sufficient to describe human daily life, since our life is made up of many chronologically ordered events, where we perform various ADLs in diverse environments while targeting a variety of objects. For example, a man drinks coffee in the living room after making coffee in the kitchen and walking into the living room. To understand what people are doing when and where in the real world, we need to recognize the sequence of combinations of *semantic concepts* (e.g., action, object and place) detected by sensors in addition to predicting each ADL label.

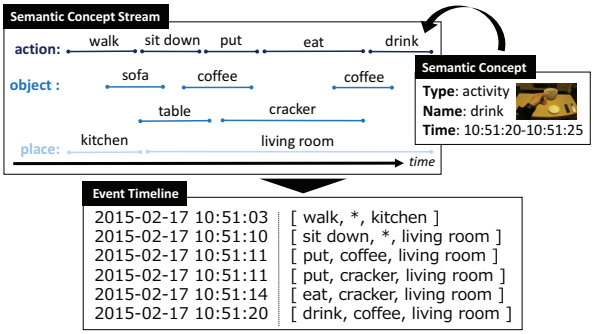

Figure 1: Illustration of generating an event timeline from a semantic concept stream.

In this paper, we propose a method to generate event timelines by generating a stereotypical event sequence of ADLs from sensor data. Figure 1 shows the process of our method when a man drinks coffee after moving to the living room. Our method translates sensor data into multiple semantic concepts of ADLs and produces a set of stereotypical sequential events with a timestamp by selecting appropriate concepts from the semantic concept stream. However, it is unknown which semantic concepts are related to each ADL, since temporal misalignment among sensors is quite frequent (Crispim-Junior et al. 2016). Moreover, there are many end-to-end approaches that translate sensor data (especially videos) of human activities into language-based descriptions (Donahue et al. 2015; Venugopalan et al. 2015b;

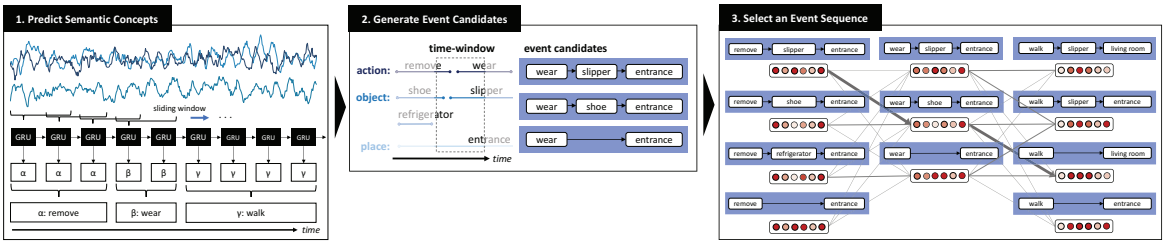

Figure 2: Overview of framework: 1. making semantic concept from sensor data of ADLs, 2. generating event candidates from semantic concept stream, and 3. selecting the most likely event sequence from event candidates using real-world properties.

2016). However, even though daily activities change over time (e.g., people put on slippers after removing shoes, rather than putting on shoes again), most existing methods do not consider such temporal variation of ADL events.

To address these problems, we used three real-world properties for generating an event timeline. First, we used temporal interactions among semantic concepts to remove objects and places unrelated to each action. Second, we used commonsense knowledge mined from a language corpus to find a possible combination of concepts in the real world. Third, we used temporal variations of ADLs to filter repetitive events. Finally, we integrated these real-world properties and generated structured events for ADLs. We assumed that by incorporating such real-world prior knowledge into the model, it would be possible to find a correct combination of concepts with a few model parameters so that the proposed method would be robust to diverse environments, in contrast to the end-to-end approaches commonly used for video captioning.

We evaluated our proposed method on a daily-activity dataset collected in a house, which contains motion and egocentric videos and human-annotated language descriptions of continuous daily activities. The experimental results show that the proposed method can generate a series of events in unseen places with high accuracy by using real-world prior knowledge such as temporal relations among concepts, commonsense knowledge from external language resources, and temporal variation of events when semantic concepts are given. Furthermore, we found that our method significantly outperforms the end-to-end deep learning approach used for video captioning.

The remainder of the paper is organized as follows. First, we present the details of our framework for generating an event timeline. Then, we provide some background information on activity recognition and video-captioning methods. Finally, we present an experimental evaluation with the ADL datasets collected in house-like settings, followed by our conclusions.

Proposed Approach

Our goal is to produce a temporally ordered event sequence of ADLs (i.e., an event timeline). To this end, the proposed method combines multiple semantic concepts made from sensor data and generates an event timeline as shown in Figure 1. We describe the proposed event timeline generation framework, consisting of three main steps, as shown in Figure 2: generating semantic concepts made from sensor data, generating candidates of events from a semantic concept stream, and selecting an event sequence based on the learned Hidden Markov Model (HMM) using three real-world properties. We introduce these steps after giving definitions of the semantic concepts and events.

Semantic Concepts and Events

We represent the real-world states as a sequence of events $\boldsymbol{e} = [e_1, \ldots, e_{|\boldsymbol{e}|}]$, where each event e is denoted as a tuple of semantic concepts $(c_1, \ldots, c_N)$. Here, each concept c is described by timestamps $c.t$ and $c.t'$, which are start and end times of the concept, concept name $c.w$ (i.e., label of the concept), and concept type $c.\hat{t} \in \{a, o, p\}$, where a is the action a subject does, o is the object he/she works on, and p is the place he/she stands. In this study, we use a tuple (c_1, c_2, c_3), where $(c_1.\hat{t} = a, c_2.\hat{t} = o, c_3.\hat{t} = p)$, as an event; in other words, c_1, c_2, and c_3 denote action, object, and place concepts. Note that object concept c_2 may be nothing, indicated by the symbol '*' in an event, when a subject does not work with any object (e.g., walking or standing). We assumed action and place concepts do not temporally overlap in the stream while object ones may overlap since people can use multiple objects at the same time. The concept type $c.\hat{t}$ specifies the values of concept name $c.w$ summarized in Table 1. Event time is defined as $e.t$, which equals the start time of action concept in event e, since each event represents ADL. For example, the event of "a man drinks coffee in the living room" at 8:31:55 in a house is denoted as e = ($c_1.w$ = *drink*, $c_2.w$ = *coffee*, $c_3.w$ = *living_room*) and $e.t$ = 8:31:55. All events $\boldsymbol{e}$ are temporally ordered by $e.t$ to represent the event timeline.

Generating Semantic Concepts

In this section, we describe generating semantic concepts for ADLs. We use the sequence-labeling approach for translating sensor data into semantic concepts of *action*, *object*, and *place* as shown in Figure 2 (left). First, we use a sliding window method for feature extraction and predict labels of sensor data in each time-window (window size is N.) Second, we use a gated recurrent unit (GRU) (Chung et al. 2014; Cho et al. 2014) for sequence labeling of sensor data. For the GRU, we assumed that each step t has an input vector x_t, a label of sensor data y_t (one-hot vector), and a hidden

state h_t. The function of the GRU is defined as:

$$\begin{aligned} z_t &= \sigma(W_z x_t + U_z h_{t-1} + b_z) \\ r_t &= \sigma(W_r x_t + U_r h_{t-1} + b_r) \\ h'_t &= \tanh(W x_t + r_t \circ U h_{t-1} + b_h) \\ h_t &= (1 - z_t) \circ h_{t-1} + z_t \circ h'_t \end{aligned}$$

where σ is a sigmoid function, $\circ$ is a Hadamard product, $W_z, W_r, W \in \mathbb{R}^{n_H \times n_I}$ and $U^{(z)}, U^{(r)}, U \in \mathbb{R}^{n_H \times n_H}$. The dimensions n_I are the size of the input vector, and the dimensions n_H are the size of the hidden vector. $b^{(z)}, b^{(r)}, b^{(h)}$ are bias terms. We refer to the above function as $h_t = GRU(x_t, h_{t-1})$. We predict labels of sensor data x_t at each window t by maximizing the conditional probability $p(y_t|x_t) = \frac{\exp(W_o h_t + b_o)}{\sum_i \exp(W_o h_i + b_o)} \cdot y_t$ of the GRU, where $W_o \in \mathbb{R}^{n_L \times n_H}$. The dimensions n_L are the size of labels. Then we merge adjacent labels into that of a semantic concept with start and end times. As a result, we can obtain temporally ordered semantic concepts (i.e., a semantic concept stream) consisting of action, object, and place concepts. Note that our framework can plug in any methods for making semantic concepts with any sensors.

Generating Candidates of an Event

In this section, we explain how to generate candidates of events by selecting relevant sets of semantic concepts from the semantic concept stream. Since the semantic concept stream consists of the unsegmented sequence of semantic concepts as shown in Figure 1, we segment it using a sliding window method and generate candidates of events from each window. We assumed each event represents an ADL (i.e., # of windows equals to # of actions). The method generates event candidates in the following manner. (i) The sliding-window method finds action concepts from the semantic concept stream, (ii) it extracts fixed-sized windows $\boldsymbol{w} = [w_1, \ldots, w_{|\boldsymbol{w}|}]$ around the start time of an action concept $c_1.t$ as denoted $w.t$, (iii) it finds time-overlapping object and place concepts within the time-window (window size is M), and (iv) it generates candidate events filling out a tuple (c_1, c_2, c_3). Figure 2 (center) shows an example of our sliding window method. For example, if the method finds an action concept *wear* from 8:31:55 to 8:31:58 within a time-window for three seconds, the method considers *wear*, *slipper*, and *shoe* as relevant object concepts, and *entrance* as a relevant place concept. As a result, multiple event candidates $e = (c_1, c_2, c_3)$, where $(c_1.w, c_2.w, c_3.w)$ = (*wear*, *slipper*, *entrance*), (*wear*, *shoe*, *entrance*), and (*wear*, *, *entrance*) are generated in a window. Then we find the most likely sequence of events over windows using real-world properties.

Learning Event Sequence

To efficiently find the most likely event sequence $\boldsymbol{e}^* = [e_1, \ldots, e_{|\boldsymbol{w}|}]$ from many candidates of events over windows, we use the hidden Markov model (HMM). The HMM is composed of the transition probability between hidden states and the emission probability between hidden and emission states. We define the hidden states of HMM as

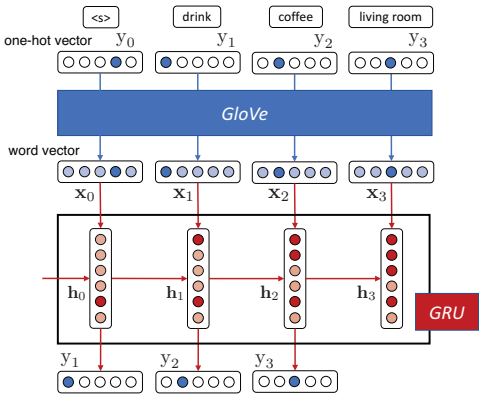

Figure 3: Event language model using GRU, which uses word embeddings by Glove as input

event e and the corresponding emissions as window w indicates event candidates. The system finds the optimal event sequence $\boldsymbol{e}^*$ from the event candidate set by maximizing the following function.

$$\underset{\boldsymbol{e}^*}{\operatorname{argmax}} \prod_{i=1}^{|\boldsymbol{w}|} p(e_i|w_i) \prod_{i=1}^{|\boldsymbol{w}|+1} p(e_i|e_{i-1}), \tag{1}$$

where $p(e_i|w_i)$ is the emission probability and $p(e_i|e_{i-1})$ is the transition probability. We assumed that the temporal property of events and the lexical property represented by a language model of events are independent of each other. Thus, we decompose $p(e|w) = p(e_t|w)p(e_l|w)$, where $p(e_t|w)$ represents the temporal relations model and $p(e_l|w)$ represents the event language model. When each window does not have $\hat{e}$, $p(e = \hat{e}|w) = 0$. $p(e_i|e_{i-1})$ can be seen as the transition between events in windows w_i and w_{i-1}. We show the details of each probability in the following sections.

Temporal Relations Model In this section, we define the temporal relations model $p(e_t|w)$, where $e_t = (c_1.t, c_2.t, c_3.t)$. It takes into account the temporal interactions of human activity related to object and place. We assumed that objects and places related to activities are observed close to the time when the action is performed. For example, when people drink a cup of coffee, they nearly simultaneously use the coffee cup. By modeling temporal relations of semantic concepts, we filter irrelevant event candidates containing unrelated objects and place concepts at each activity. We model temporal relations as follows.

$$p(e_t|w) \propto \sum_{i=1}^{|e|} \exp(-\alpha|c_i.t - w.t|) \tag{2}$$

where $w.t$ is time of window w and $c.t$ denotes the start time of each semantic concept of event e. We use $p(e_t|w)$ as a part of the emission probability of the proposed HMM model.

Event Language Model To estimate the likelihood of concept combinations, we use a language model learned by

a language corpus. We assumed that language texts in the corpus describe real-world commonsense knowledge, which represents the likelihood of events that will occur in the real world. For example, people drink coffee rather than a sandwich cookie. Moreover, people drink coffee in the living room rather than in the toilet room. To represent such commonsense knowledge in the real world, we use the language model of an event $p(e_l|w)$, where $e_l = (c_1.w, c_2.w, c_3.w)$. For this language model, we also use the GRU, that can represent the word sequence of an arbitrary-length. When using the GRU as a language model, the k-th word probability is defined as $p(y_k|x_k) = \frac{\exp(W_{\hat{o}}h_k + b_{\hat{o}})}{\sum_i \exp(W_{\hat{o}}h_i + b_{\hat{o}})} \cdot y_k$ where y_k is a one-hot vector of k-th word, a hidden state $h_k = GRU(x_k, h_{k-1})$, given a k-th input word vector x_k. In addition, we use the vector space representations of words learned by GloVe (Pennington, Socher, and Manning 2014) as input of the GRU language model for addressing the case when the semantic concept names do not appear in the training corpus. We define x_k as the Glove word vector of word $c_i.w$ in an event. We use the joint probability of

$$p(e_l|w) = \prod_{k=1}^{|e|} p(y_k|x_k) \tag{3}$$

when reading the word embeddings of the last word $c_3.w$ as the likelihood of the GRU language model. Figure 3 shows the GRU language model with the word embeddings of Glove.

Event Sequence Model In the real world, our ADLs gradually change over time. For example, people put on slippers after removing shoes, rather than putting on shoes again. We incorporate this temporal variation of events into the HMM model. We assumed difference of vector representations of events describes temporal variation of ADL events. To this end, we define the transition probability as dissimilarity between vectors of events e_i and e_{i-1}. We have

$$p(e_i|e_{i-1}) \propto \exp(-\beta|q_i - q_{i-1}|), \tag{4}$$

where q_i is the final hidden state of a GRU when reading the words in the event, which is a vector representation of event in a window w_i. Figure 2 (right) illustrates the event transition, which shows that the path of (*remove*, *shoe*, *entrance*) $\rightarrow$ (*wear*, *slipper*, *entrance*) $\rightarrow$ (*walk*, *, *living_room*), is more likely to occur than that of (*remove*, *shoe*, *entrance*) $\rightarrow$ (*wear*, *shoe*, *entrance*) $\rightarrow$ (*walk*, *slipper*, *entrance*), since the hidden vector of event (*wear*, *slipper*, *entrance*) is semantically dissimilar to that of (*remove*, *shoe*, *entrance*) rather than that of (*wear*, *shoe*, *entrance*).

Inference of Event Sequence

There are several paths through the hidden states that represent an event sequence. Figure 2 (right) shows the lattice of our HMM that represents a sequence of events. Each column shows a window holding event candidates. Our model is an instance of an HMM, and therefore the computation of marginals is tractable. To efficiently find the most likely sequence for events, we apply a Viterbi algorithm (Rabiner 1989) using the normalized emission $p(e_i|w_i)$ and transition probabilities $p(e_i|e_{i-1})$ of events over windows.

Related Work

The purpose of this work is to generate language descriptions about activities of daily living by using sensor data (e.g., motion and video data). Here, we show related work on activity recognition and video captioning.

To recognize human activities of daily life, many approaches used classification or sequence labeling of indoor activities with pervasive (Buettner et al. 2009; Tapia, Intille, and Larson 2004; Van Kasteren et al. 2008) and wearable motion sensors (Bao and Intille 2004; Hammerla, Halloran, and Plötz 2016). Recently, many vision-based activity recognition methods have been proposed using a head-mounted wearable camera (Ramanan 2012; Ma, Fan, and Kitani 2016) and a wrist-worn camera (Maekawa et al. 2010; Ohnishi et al. 2016) for capturing everyday activities and their related objects in diverse home environments. We also used wearable sensors for making semantic concepts with the same motivation. However, these existing methods focus on predicting activity labels from sensor data. In contrast, in this paper, we mainly focus on generating a sequence of ADL events (i.e., an event timeline) that consists of a combination of multiple semantic concepts in addition to recognizing activities. Note that our method can plug in any activity recognition methods for making action concepts.

There has been significant research interest in generating language description from videos, which is called video captioning. Many approaches used sequence labeling techniques such as CRF (Regneri et al. 2013), statistical machine translation technique (Rohrbach et al. 2013), and the end-to-end deep learning approach, which encodes time-varying visual input with convolutional neural networks (CNNs) and decodes a variable-length sentence with long-term/short-term memory (Donahue et al. 2015) or generates multiple sentences using hierarchical Recurrent Neural Networks (RNNs) (Yu et al. 2016). However, they used the TACoS Multi-Level corpus, which involves human-activity videos captured only in a kitchen scenario with fixed camera settings. In contrast, our dataset used in this paper has been collected by wearable sensors to evaluate methods with sensor data in diverse places. Moreover, our methods can generate language descriptions of various activities conducted in diverse places by using real-world properties. The recent study of video captioning for YouTube videos and movie clips uses the end-to-end approach based on deep learning (Venugopalan et al. 2015c; 2015a; 2016; Krishna et al. 2017). However, the end-to-end approaches do not work when testing in an environment different from the training environments, since the end-to-end deep learning methods directly learn the relationships between video data and language descriptions. In contrast, our proposed method can accurately generate an event timeline in unseen places by using real-world properties when semantic concepts from sensor data are given.

Experiments

Datasets

We evaluated our proposed method using datasets of ADLs, which are manually annotated with structured event descrip-

Table 1: List of names of semantic concepts used for actions, objects, and places.

action: a	object: o	place: p
'brush' 'close' 'drink' 'eat' 'flip' 'flush' 'gargle' 'hold' 'make' 'open' 'pour' 'put' 'read' 'remove' 'sit_down' 'sleep' 'stand_up' 'stir' 'throw_away' 'turn_off' 'unroll' 'walk' 'wash' 'wash_face' 'watch' 'wear' 'wipe'	'air_conditioner' 'bed' 'book' 'bottle' 'butter_knife' 'coffee' 'cracker' 'cream_cheese' 'cup' 'dishwasher' 'duster' 'faucet' 'floor' 'food_package' 'food_container' 'glass' 'hand' 'hot_water' 'mop' 'plate' 'pot' 'refrigerator' 'remote_control' 'shelf' 'shoe' 'sink' 'slipper' 'sofa' 'sponge' 'spoon' 'switch' 'table' 'television' 'toilet' 'toilet_paper' 'toothbrush' 'towel' 'water' 'wristwatch'	'bathroom' 'bedroom' 'entrance' 'kitchen' 'living_room' 'washroom'

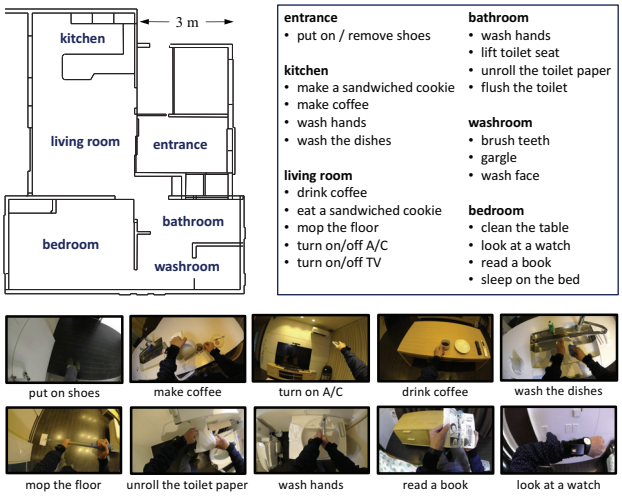

Figure 4: Layout of experiments, list of daily activities in different places, and experimental places captured by wearable camera.

tions of the ADLs that subjects performed in a house. This dataset has been used in egocentric video retrieval with gesture motions (Miyanishi et al. 2016). The dataset consists of motion signals (e.g., acceleration and gyro) and first-person vision videos captured by a head-mounted wearable camera that enables us to capture various ADLs in diverse places. To collect real-world data, 8 subjects wearing motion sensors and a wearable camera performed continuous 20 ADLs in 6 different places 10 times (i.e., in total 10 sessions) according to the instructions on a worksheet under the semi-naturalistic collection protocol (Bao and Intille 2004) to collect more variable behavior data than in a laboratory setting. For example, the subject sat on the sofa and drank a cup of coffee while watching TV in the living room, and then he moved to the kitchen to wash dishes. Total motion signals and video length are about 17 hours. A single session averaged 10.86 minutes over the subjects. Figure 4 shows the layout of the house and a list of the 20 activities the subject performs in the different places.

To evaluate the event sequence generated by our proposed method, we annotated event descriptions to this ADL dataset. We had two annotators manually label semantic concepts of the 27 activities, 39 objects, and 6 places shown in Table 1 by watching 17 hours of videos from wearable cameras attached to the heads of subjects. Then, we instructed the six annotators to manually re-order labeled concepts for making correct events that describe real-world events using crowd-sourcing services. For example, annotators first selected an action concept and then object and place concepts related to the selected action concept. Then they re-ordered them to make event descriptions such as (*drink*, *coffee*, *living_room*) when drink (activity), coffee (object), living_room (place) concepts were nearly simultaneously observed. In total, we obtained 11,501 action concepts, 13,087 object concepts, 931 place concepts, and 11,280 descriptions of ADL events with 550 unique events.

Experimental Settings

Generating Semantic Concepts We made semantic concepts from the signals of wearable sensors. For feature extraction from motion signals and videos, we followed the past work (Miyanishi et al. 2016). For the motion-feature extraction, we used acceleration and gyro signals and applied a short-time Fourier transform, where the window width was 75 samples (3 sec when using 25-Hz data) by shifting one sample. We standardized motion features using the mean and variance after downsampling the transformed signals from 25 to 10 Hz to align the sampling rate to the video features. For the video feature extraction, we used a sliding window method to obtain the video features, which consist of image features extracted by CNNs with a pre-trained model of VGG (Simonyan and Zisserman 2015). We used the activations in the second-to-last fully connected layer as image features and then applied PCA to the image features and reduced the dimensions from 4,096 to 250. Then, we also down-sampled all image features from 29.97 to 10 Hz by a rolling mean of the time intervals. We combined image feature vectors within a time-window (window size is 3 sec) for making video features and standardized the video features with the mean and the variance.

We made all semantic concepts using the combined motion features and video features. To make concepts, we predicted the labels of sensor data using the GRU with the sliding-window method. We set the window size to N=3 sec. We trained our GRU using the Adam optimizer (Kingma and Ba 2015), with a learning rate of 0.001 and a batch size of 32. We set hidden dimensions of GRU 300 initialized with a Normal distribution $\mathcal{N}(0, 0.01)$. Its parameters were optimized to get the best validation performance in training runs for up to 30 epochs. We used the cross entropy

loss for a multi-class label of action and place. We used a multi-label one-versus-all loss based on the max-entropy for multi-labeling of object labels. Then, we predicted labels of semantic concepts with leave-one-cross-session training, which trains a model on the nine sessions, and tested it with another session's data. The classification performance (F1-score) of action, object, place labels were 0.7479, 0.6474, and 0.9855, respectively. Then, we merged the adjacent labels and made concepts. As a result, we obtained 14,444 action concepts, 34,923 object concepts, and 1,113 place labels. We refer to these two types of concepts, that is, manually labeled semantic concepts and predicted ones, as TrueConcept and PredConcept, respectively. Note that we used the same TrueConcept and PredConcept for all methods to generate event timelines. Thus we fairly compare the event generation performance of methods when the semantic concepts are given.

Generating Event Timeline We generated a sequence of events using semantic concepts made from sensor data by using the proposed method. To learn the event language model, we used language descriptions of the Montreal Video Annotation Dataset (M-VAD)[1], that describes real-world events. We applied the following preprocessing to obtain the combinations of action, object, and place concepts. For extracting verbs and their dependencies, we used the Stanford dependency parser using Neural Networks (Chen and Manning 2014). We extracted verbs with "nsubj" and "xcomp" (clausal complement of a verb) nodes to find actions. Then we extracted their verb dependency with "dobj" nodes to find target objects of actions. Moreover, we used "nmod" (noun dependency) nodes related to target objects to find places. The event component vocabulary comprised the 4,000 most common verbs, 6,000 most common nouns of verb dependencies, and 6,000 most common nouns of prior noun dependencies. Our training set for the language model was a total of 51,564 events, and 6,871 events were used for validation.

Our event language model by GRU used the word embeddings by Glove. To learn the word embeddings, we used the 400,000 most common words in English-language Wikipedia. All other words were replaced with an unknown $\langle$UNK$\rangle$ token. We set hidden dimensions of Glove 300 following (Pennington, Socher, and Manning 2014). For learning event language model, we use the same parameter of GRU of making semantic concepts. Furthermore, we optimized its parameters to get the best validation performance in training runs for up to 10 epochs.

The event generation method needs to generalize the diverse places. We tuned a few hyper-parameters the window-size M for generating candidates of the event, α for the temporal model, and β for the event sequence model among candidates $M = \{1, 2, 3, 4, 5\}$, $\alpha = \{0, 0.2, 0.4, 0.6, 0.8\}$ and $\beta = \{0, 0.2, 0.4, 0.6, 0.8\}$. We used leave-one-place-out cross-validation, optimized for the best performance in Bleu score (as described later), on the validation data of five places (not including a target place). Finally, we tested the method with the target place dataset for each subject. By using the cross-place-validation, we can see how the proposed methods robustly improve performance in diverse places comparing to baselines.

[1]https://mila.quebec/en/publications/public-datasets/m-vad/

Methods

Our proposed method first generates event candidates by selecting concepts from the semantic concept stream according to the log scores of Eq. (1), which is the sum of log probabilities of the temporal relations model in Eq. (2), the event language model in Eq. (3), and the event sequence model in Eq. (4). To compare the effectiveness of each component in the proposed approach, we prepared three methods: GRU-Lang, GRU-Lang+Time, and GRU-Lang+Time+Context. GRU-Lang uses the score of the event language model for event timeline generation. GRU-Lang+Time uses the score of the temporal model in addition to the score of GRU-Lang. GRU-Lang+Time+Context selects event sequences based on HMM using the event sequence model in addition to the score of GRU-Lang+Time. Moreover, we prepared several baselines: Random, Unigram, Bigram, S2VT, and S2VT Ranking. Random randomly selects events among event candidates in each window to generate an event sequence. Unigram and Bigram rank events over windows using the log-probability of unigram and bigram on the M-VAD corpus. By comparing Unigram and Bigram to GRU-Lang, we can see the strength of the GRU language model. To compare standard end-to-end video-captioning methods, we applied S2VT (Venugopalan et al. 2015a), which encodes a sensor data using one GRU and decodes events using another GRU over each window holding event candidates. We set the parameters of S2VT following the past work (Venugopalan et al. 2015a). We also prepared S2VT-Ranking to rank events among event candidates in each window according to the word sequence likelihood of GRU in S2VT. Estimating the likelihood of word sequence by S2VT-Ranking is almost same to GRU-Lang, but S2VT-Ranking use without use of external language resource. We used the leave-one-cross-place-validation for test data in an environment different from ones used for training or tuning their parameters. We assumed the S2VT and S2VT-Ranking tend to overly fit the data in a learned environment in contrast to our method since the end-to-end approach directly learn the relationship between sensor data and language descriptions even if semantic concepts are given.

Event Generation Performance

In this section, we evaluate the performance of our proposed method and compare our methods to baselines that use word statistics of external language resources and the end-to-end caption-generation approach. These baselines do not incorporate real-world properties such as temporal relationships between concepts and temporal variation of events.

To generate events, we used both TrueConcept and PredConcept. By using TrueConcept, we can see the pure performance of event timeline generation. Using PredConcept assumed a more practical situation. This condition uses sensor data to automatically predict semantic concepts that rep-

Table 2: Performances of event timeline generation by different methods when using TrueConcept and PredConcept.

	with TrueConcept					with PredConcept				
Method	Bleu@1	Bleu@2	Bleu@3	CIDEr	METEOR	Bleu@1	Bleu@2	Bleu@3	CIDEr	METEOR
Random	0.8051	0.5648	0.5098	3.2954	0.4107	0.7163	0.4545	0.3627	2.4653	0.3537
Unigram	0.7601	0.4811	0.3865	2.9660	0.3786	0.6790	0.4086	0.2710	2.3818	0.3377
Bigram	0.8111	0.6220	0.5660	3.8151	0.4476	0.6925	0.4701	0.3814	2.6310	0.3603
S2VT	0.0955	0.0232	0.0023	0.2045	0.0505	0.0983	0.0425	0.0048	0.2753	0.0567
S2VT-Ranking	0.7708	0.5843	0.4437	3.3487	0.4005	0.6839	0.4862	0.3120	2.6801	0.3550
GRU-Lang (ours)	0.8392	0.6475	0.5911	3.9751	0.4589	0.7115	0.4739	0.3849	2.6507	0.3587
+ Time	0.8510	0.6769	0.6279	4.1519	0.4680	0.7303	0.5092	0.4276	2.8646	0.3755
+ Time + Context	**0.8546**	**0.7126**	**0.6768**	**4.3306**	**0.4793**	**0.7450**	**0.5596**	**0.4776**	**3.1421**	**0.3943**

resent real-world states. It tells us whether the proposed method works even under such a practical situation.

Evaluation Measure To evaluate the performance of generating a sequence of events, we used common evaluation metrics: Bleu, METEOR, and CIDEr. These metrics are used for evaluating image and video captioning (Fang et al. 2015; Krishna et al. 2017), and we calculated them using the codes of MS COCO caption evaluation[2]. We assumed that the real-world properties improve the performance of describing a continuous event sequence. We then compared the generated events to the human-annotated true events made using Bleu, METEOR, and CIDEr scores. Note that when using PredConcept, we found nearest predicted events to true ones based on timestamps of event and calculated evaluation measures over paired events. Here, a higher metric score is better.

Experimental Results We investigated how well our model performs when generating event timelines. Table 2 shows the overall results of a baseline and our proposed methods when using both TrueConcept and PredConcept. The results when using TrueConcept in Table 2 (left) show that GRU-Lang performs significantly better than the Unigram and Bigram over all evaluation metrics, even though these methods use the same language corpus to build their language model. This indicates that the language model used in GRU-Lang could more accurately represent real-world knowledge by using an external language source, since GRU models the word sequence of an arbitrary-length in comparison to a standard language model. In addition, GRU-Lang significantly outperformed S2VT and S2VT-Ranking across all evaluation metrics, since the end-to-end approaches overfit the sensor data collected in the specific environment and could not predict events in unseen environments even if semantic concepts are given. This indicates that we need to use the external resource for modeling real-world knowledge to generalize event timeline generation methods applicable to the diverse environments. The results when using TrueConcept show that the proposed GRU-Lang, GRU-Lang+Time, and GRU-Lang+Time+Context significantly outperformed the baselines. This suggests that the modeling of real-world properties is highly effective for generating an event sequence of ADLs. In particular, GRU-Lang+Time performs better than GRU-Lang, indicating that considering temporal relations of semantic concepts is essential to generating correct events in the real world. The finding that GRU-Lang+Time+Context outperforms GRU-Lang+Time shows that modeling an event sequence improves the performance of event timeline generation. Moreover, GRU-Lang+Time+Context outperforming GRU-Lang and GRU-Lang+Time suggests that the integration of various real-world properties such as temporal relations, real-world commonsense knowledge, and the variation of events is more effective than using only one or two properties.

The performance when using PredConcept decreases compared to that when using TrueConcept, since the prediction of semantic concepts is not perfect. Table 2 (right) shows that the proposed GRU-Lang outperformed Unigram; however, the performance of GRU-Lang and Bigram is almost the same. This indicates GRU-Lang is slightly weak when using event candidates made with PredConcept because predicted concepts are sometimes incorrect. However, GRU-Lang+Time improved the performance of event generation and outperformed Unigram, Bigram, and GRU-Lang, suggesting that incorporating temporal properties is effective for building an event timeline in practical situations. GRU-Lang+Time+Context also outperformed GRU-Lang and GRU-Lang+Time, similar to when using TrueConcept. It appears that incorporating a variation of events is also effective for selecting an accurate sequence of events based on semantic concepts predicted by sensor data. The result suggests that we can generate a more correct event timeline in practical situations by incorporating all real-world properties.

Parameter Sensitivity In this section, we report the sensitivity of the proposed methods' hyper parameters, which we tuned with the leave-one-cross-place validation. We demonstrate in Figure 5 how the values of Bleu for GRU-Lang+Time change with different α parameters. We assumed that the penalty of the distance among timestamps of semantic concepts filters incorrect events because people do various activities under time constraints. As the parameter α increases, the Bleu scores of GRU-Lang+Time also increase when using both TrueConcept and PredConcept. As the parameter α takes a negative value or zero, the Bleu scores do not outperform when α takes positive. This indicates that the event including temporally close semantic

[2]https://github.com/tylin/coco-caption

Table 3: Sequence of events obtained by proposed method with TrueConcept.

Timestamp	True Event	GRU-Lang	GRU-Lang+Time	GRU-Lang+Time+Context
2015-02-20 11:59:30	walk, *, kitchen	walk, faucet, kitchen	walk, *, kitchen	walk, *, kitchen
2015-02-20 11:59:33	flip, faucet, kitchen	flip, faucet, kitchen	flip, faucet, kitchen	flip, faucet, kitchen
2015-02-20 11:59:36	wash, hand, kitchen	wash, hand, kitchen	wash, water, kitchen	wash, hand, kitchen
2015-02-20 11:59:40	turn_off, faucet, kitchen	turn_off, faucet, kitchen	turn_off, faucet, kitchen	turn_off, faucet, kitchen
2015-02-20 11:59:42	wipe, hand, kitchen	wipe, hand, kitchen	wipe, hand, kitchen	wipe, hand, kitchen
2015-02-20 11:59:51	walk, *, kitchen	walk, bottle, kitchen	walk, bottle, kitchen	walk, *, kitchen
2015-02-20 11:59:53	hold, bottle, kitchen	hold, bottle, kitchen	hold, bottle, kitchen	hold, bottle, kitchen
2015-02-20 11:59:55	walk, *, kitchen	walk, bottle, kitchen	walk, bottle, kitchen	walk, *, kitchen
2015-02-20 12:00:00	put, bottle, kitchen	put, bottle, kitchen	put, bottle, kitchen	put, bottle, kitchen
2015-02-20 12:00:03	close, shelf, kitchen	close, *, kitchen	close, *, kitchen	close, *, kitchen

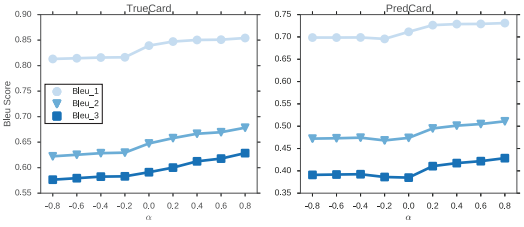

Figure 5: Sensitivity to GRU-Lang+Time hyper parameter. The x-axis shows the α value of the time-aware model; the y-axis shows Bleu score.

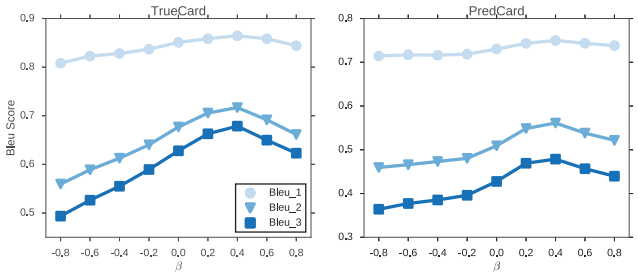

Figure 6: Sensitivity to the GRU-Lang+Time+Context hyper parameter. The x-axis shows the context-aware model's value of β; the y-axis shows the value of Bleu.

concepts is correct supporting our assumption that temporal relations among concepts provide an important factor in filtering incorrect events, helping to improve the performance of event generation.

Figure 6 shows the sensitivity of parameter β in GRU-Lang+Time+Context, which controls the variations of events. If β is small, the method tends to generate semantically similar events. In contrast, if β is large, the method outputs different events before and after this parameter is applied. We assumed the optimal value of β is positive, since ADL events in the real world slightly change. According to the curves corresponding to Bleu scores, the performance of GRU-Lang+Time +Context increased until β=0.4 and then decreased over all Bleu scores when using both TrueConcept and PredConcept. Moreover, as the parameter β takes a negative value, the Bleu scores decrease. These results support our assumption and suggest that incorporating a variation of events into the model plays an important role in generating an accurate event timeline.

Qualitative Analysis To understand what is predicted by the proposed event timeline generation methods, we report the sequence of events using TrueConcept to see the pure performance of event timeline generations. Table 3 shows the sequence of events for the preparation of making coffee performed in the kitchen. According to this table, GRU-Lang+Time using temporal properties can predict more accurate events than can GRU-Lang. For example, GRU-Lang+Time predicts a true event "(walk, *, kitchen)" at 11:59:30, while GRU-Lang incorrectly outputs a wrong event "(walk, faucet, kitchen)." However, GRU-Lang+Time could not capture the event transitions, and thus it produced a wrong event sequence such as "(hold, bottle, kitchen)"→"(walk, bottle, kitchen)"→"(put, bottle, kitchen)" from 11:59:53 to 12:00:00. In contrast, by incorporating a temporal variation of events into the model, GRU-Lang+Time+Context can generate a more accurate sequence of events than can GRU-Lang+Time. The method outputs an accurate event sequence for carrying the object to another place, such as "(hold, bottle, kitchen)"→"(walk, *, kitchen)"→"(put, bottle, kitchen)."

Conclusion

This paper showed a novel framework for generating an event timeline of ADLs using sensor data. The model uses real-world properties such as the temporal relationship of concepts, commonsense knowledge, and the temporal variation of events. The experiment using ADL dataset show that modeling real-world properties significantly improves the performance of generating an event timeline.

Acknowledgments

This work was supported by CREST and SICORP from JST, and JSPS KAKENHI 16K21718.

References

Bao, L., and Intille, S. S. 2004. Activity recognition from user-annotated acceleration data. In *Pervasive*, 1–17.

Buettner, M.; Prasad, R.; Philipose, M.; and Wetherall, D. 2009. Recognizing daily activities with RFID-based sensors. In *UbiComp*, 51–60.

Castro, D.; Hickson, S.; Bettadapura, V.; Thomaz, E.; Abowd, G.; Christensen, H.; and Essa, I. 2015. Predicting daily activities from egocentric images using deep learning. In *ISWC*, 75–82.

Chen, D., and Manning, C. D. 2014. A fast and accurate dependency parser using neural networks. In *EMNLP*, 740–750.

Cho, K.; van Merriënboer, B.; Bahdanau, D.; and Bengio, Y. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *SSST-8*.

Chung, J.; Gülçehre, Ç.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Deep Learning and Representation Learning Workshop*.

Crispim-Junior, C. F.; Buso, V.; Avgerinakis, K.; Meditskos, G.; Briassouli, A.; Benois-Pineau, J.; Kompatsiaris, I. Y.; and Bremond, F. 2016. Semantic event fusion of different visual modality concepts for activity recognition. *IEEE transactions on pattern analysis and machine intelligence* 38(8):1598–1611.

Donahue, J.; Hendricks, L. A.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; and Darrell, T. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2625–2634.

Fang, H.; Gupta, S.; Iandola, F.; Srivastava, R. K.; Deng, L.; Dollár, P.; Gao, J.; He, X.; Mitchell, M.; Platt, J. C.; et al. 2015. From captions to visual concepts and back. In *CVPR*, 1473–1482.

Hammerla, N. Y.; Halloran, S.; and Plötz, T. 2016. Deep, convolutional, and recurrent models for human activity recognition using wearables. In *IJCAI*, 1533–1540.

Inoue, S.; Ueda, N.; Nohara, Y.; and Nakashima, N. 2015. Mobile activity recognition for a whole day: Recognizing real nursing activities with big dataset. In *UbiComp*, 1269–1280.

Kingma, D., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Krishna, R.; Hata, K.; Ren, F.; Fei-Fei, L.; and Niebles, J. C. 2017. Dense-captioning events in videos. In *ICCV*.

Ma, M.; Fan, H.; and Kitani, K. M. 2016. Going deeper into first-person activity recognition. In *CVPR*, 1894–1903.

Maekawa, T.; Yanagisawa, Y.; Kishino, Y.; Ishiguro, K.; Kamei, K.; Sakurai, Y.; and Okadome, T. 2010. Object-based activity recognition with heterogeneous sensors on wrist. In *Pervasive*, 246–264.

Meditskos, G.; Kontopoulos, E.; and Kompatsiaris, I. 2014. Knowledge-driven activity recognition and segmentation using context connections. In *ISWC*, 260–275. Springer.

Miyanishi, T.; Hirayama, J.-i.; Maekawa, T.; Kong, Q.; Moriya, H.; and Suyama, T. 2016. Egocentric video search via physical interactions. In *AAAI*, 330–336.

Ohnishi, K.; Kanehira, A.; Kanezaki, A.; and Harada, T. 2016. Recognizing activities of daily living with a wrist-mounted camera. In *CVPR*.

Ordóñez, F. J., and Roggen, D. 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* 16(1):115.

Patterson, D. J.; Fox, D.; Kautz, H.; and Philipose, M. 2005. Fine-grained activity recognition by aggregating abstract object usage. In *ISWC*, 44–51.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*, 1532–1543.

Rabiner, L. R. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *IEEE* 77(2):257–286.

Ramanan, D. 2012. Detecting activities of daily living in first-person camera views. In *CVPR*, 2847–2854.

Rashidi, P., and Cook, D. J. 2010. Mining sensor streams for discovering human activity patterns over time. In *ICDM*, 431–440.

Regneri, M.; Rohrbach, M.; Wetzel, D.; Thater, S.; Schiele, B.; and Pinkal, M. 2013. Grounding action descriptions in videos. *TACL* 1:25–36.

Rohrbach, M.; Qiu, W.; Titov, I.; Thater, S.; Pinkal, M.; and Schiele, B. 2013. Translating video content to natural language descriptions. In *ICCV*, 433–440.

Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Tapia, E. M.; Intille, S. S.; and Larson, K. 2004. Activity recognition in the home using simple and ubiquitous sensors. In *Pervasive*, volume 4, 158–175.

Van Kasteren, T.; Noulas, A.; Englebienne, G.; and Kröse, B. 2008. Accurate activity recognition in a home setting. In *UbiComp*, 1–9. ACM.

Venugopalan, S.; Rohrbach, M.; Donahue, J.; Mooney, R.; Darrell, T.; and Saenko, K. 2015a. Sequence to sequence - video to text. In *ICCV*.

Venugopalan, S.; Rohrbach, M.; Donahue, J.; Mooney, R.; Darrell, T.; and Saenko, K. 2015b. Sequence to sequence-video to text. In *ICCV*, 4534–4542.

Venugopalan, S.; Xu, H.; Donahue, J.; Rohrbach, M.; Mooney, R.; and Saenko, K. 2015c. Translating videos to natural language using deep recurrent neural networks. *NAACL-HLT*.

Venugopalan, S.; Hendricks, L. A.; Mooney, R.; and Saenko, K. 2016. Improving lstm-based video description with linguistic knowledge mined from text. In *EMNLP*.

Yu, H.; Wang, J.; Huang, Z.; Yang, Y.; and Xu, W. 2016. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*.