PBG at the NTCIR-13 Lifelog-2 LAT, LSAT, and LEST Tasks

Shuhei Yamamoto
NTT Service Evolution Laboratories, Japan
yamamoto.shuhei@lab.ntt.co.jp

Takuya Nishimura
NTT Service Evolution Laboratories, Japan
nishimura.takuya@lab.ntt.co.jp

Yasunori Akagi
NTT Service Evolution Laboratories, Japan
akagi.yasunori@lab.ntt.co.jp

Yoshiaki Takimoto
NTT Service Evolution Laboratories, Japan
takimoto.yoshiaki@lab.ntt.co.jp

Takafumi Inoue
NTT Service Evolution Laboratories, Japan
inoue.takafumi@lab.ntt.co.jp

Hiroyuki Toda
NTT Service Evolution Laboratories, Japan
toda.hiroyuki@lab.ntt.co.jp

ABSTRACT

In this paper, the participation of the PBG research team in the NTCIR-13 Lifelog LAT, LSAT, and LEST tasks is described. In common with these three subtasks, our team focuses on both images and locations and analyzes them with visual and location indexing methods. The results obtained demonstrated outstanding performance, and we clarified effective features for solving lifelog tasks.

Team Name

PBG

Subtasks

Lifelog Annotation Task (English)
Lifelog Semantic Access Task (English)
Lifelog Event Segmentation Task (English)

Keywords

ImageNet, Places, D-star, DBSCAN, DNN, Gated CNN

1. INTRODUCTION

The PBG research team of NTT Service Evolution Laboratories in Japan participated in the Lifelog Annotation (LAT), Lifelog Semantic Access (LSAT), and Lifelog Event Segmentation (LEST) subtasks of the NTCIR-13 Lifelog-2 Task [4][1]. Life-logging sensors have become more and more popular, and people naturally wear them in the form of smartwatches, smartphones, and so on. Lifelog data is composed of multimodal information such as images, body metrics, e.g., heart rates, calories, and skin temperature, and GPS data, and they are used to effectively remind users' important information and improve their daily lives. However, the amount of such lifelog data increases as time proceeds, and manually finding useful information is difficult from a large amount of data. Solving this lifelog task is important for automatically indexing daily life.

The common problem with LAT, LSAT, and LEST is understanding user activity and context from lifelog data. For this task, although we can use a large amount of a specific user's lifelog data with various modals, we cannot directly obtain semantic information. Our team focuses on both images and locations because we think that the most important point for understanding user activity and context in daily life is what users are doing what, where, and when. In this paper, to obtain visual semantics from images, we apply visual indexing methods using deep neural networks trained on ImageNet and Places and detect a number of people by using OpenCV[2] library. For indexing location information on each user, we adopt D-star [8] and DBSCAN [3], each of which is known to be effective in analyzing moving trajectories.

The remainder of our paper is organized as follows. In Section 2, we describe visual indexing methods based on the public datasets of ImageNet and Places. In Section 3, we introduce a location indexing algorithm using D-star and DBSCAN. In Section 4, Section 5, and Section 6, we explain our approaches for solving each subtask and analyze official results. We conclude the paper by briefly describing future work in Section 7.

2. VISUAL INDEXING

The three subtasks, LAT, LSAT, and LEST, are required to understand a user's situation each minute. Although lifelog images provide much information for understanding a situation, they are not direct. Therefore, we estimate semantic visual labels for images from three aspects of objects, scenes, and number of people, each of which can aid in predicting user action, context, and social relation from each image.

For object recognition, we apply two deep neural network (DNN) models using GoogLeNet [12] and AlexNet [7], trained on the ImageNet dataset of ILSVRC2012 [9]. Also, for scene recognition, we use four DNN models using GoogLeNet, AlexNet, VGG [11], and ResNet [5], trained on the Places365 dataset [16]. These models were provided in the *Caffe* framework [6] via github pages[3][4]. To take advantage of several categories of features, we use the last output layer of each network, i.e., 1000 and 365 visual concepts of ImageNet and Places365, respectively.

For detecting the number of people in each image, our team uses a pedestrian detection function that uses HOG feature values with OpenCV. We get three values for the number of people by using three threshold values in the function. The values were chosen to be 1.0, 1.5 and 2.0 on the basis of our preliminary experimentation. Other parameters were set to the default values in OpenCV.

[1]http://ntcir-lifelog.computing.dcu.ie/

[2]http://opencv.org

[3]https://github.com/BVLC/caffe/tree/master/models

[4]https://github.com/CSAILVision/places365

Concept label	computer, desk, document, hand, indoor, person
ImageNet (GoogLeNet)	notebook computer: 0.557, laptop computer: 0.261, monitor: 0.043, ⋯
Places (GoogLeNet)	office: 0.191, computer room: 0.110, hospital room: 0.094, ⋯

Figure 1: Example of visual indexing result

Algorithm 1 D-Star

for each point p_i in $\mathbf{T}$ **do**
  Push p_i in sliding window.
  Create empty neighborhood $\mathbf{N}(p_i) = \Phi$.
  for each point p_i in sliding window **do**
    if $d(p_i, p_j) \leq \epsilon$ **then**
      Add p_i to $\mathbf{N}(p_j)$.
      Add p_j to $\mathbf{N}(p_i)$.
    end if
  end for
  Shift p_{i-q} from sliding window.
  if $|s(\mathbf{N}(p_{i-q}))| \geq m_{time}$ **then**
    if p_{i-q} is *duration-joinable* to an existing cluster **then**
      Merge $\mathbf{N}(p_{i-q})$ and all its duration-joinable clusters.
    else
      Create a new cluster based on $\mathbf{N}(p_{i-q})$.
    end if
  end if
end for
return clusters $\{\mathbf{C} \mid |s(\mathbf{C})| \geq T_{stay}\}$ as stay points.

With the above processes, our visual features extracted from each image consisted of 3,463 dimensions: ImageNet (1,000d) × 2 DNNs + Places365 (365d) × 4 DNNs + the number of people (3d). An example of our visual indexing result is shown in Fig. 1. This example also shows concept labels that were provided in the phase 2 dataset by the task organizers. ImageNet and Places estimated "comptuter" and "office" with a high probability value.

3. LOCATION INDEXING

A GPS trajectory is hard to use directly to understand user behavior. As preprocessing, we adopt stay point detection [15] and important location detection [14].

As stay point detection, we adopt the **D-Star** algorithm (a duration-based stay region extraction algorithm) [8]. It is based on the density-based algorithm and extended for considering duration. For each point p_i which in a trajectory, we extract the neighborhood of p_i as $\mathbf{N}(p_i) = \{p_j \in \mathbf{T} \mid d(p_i, p_j) \leq \epsilon \wedge |i - j| \leq q\}$. We call a point p_i a "core point" if the duration of its neighborhood $\mathbf{N}(p_i)$ is longer than the time threshold m_{time}. After that, we merge each pair of the neighborhoods of core points if these neighborhoods overlap. As a result, the D-Star algorithm outputs all merged result as a set of stay points. Algorithm 1 shows the algorithm of D-Star.

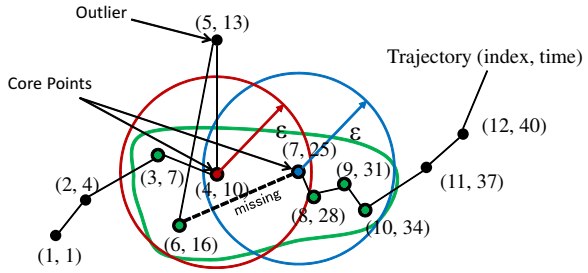

Figure 2: Example of cluster determined by D-Star for distance threshold ϵ, sliding window size $q = 3$, and duration threshold for core point $m_{time} = 15$. Point p_i is described as (i, t_i) in this figure.

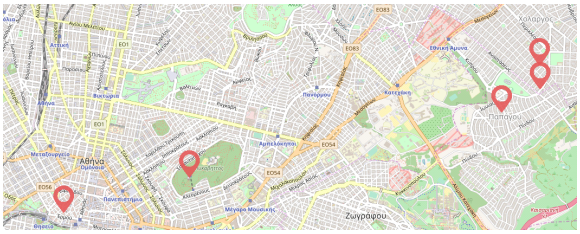

Figure 3: Important locations detected by our approach from GPS data of user 2. (c)OpenStreetMap contributors

Figure 2 shows an example of a cluster of two duration-joinable core points for a maximum distance threshold ϵ, sliding windows size $q = 3$, and minimum duration threshold for a core point $m_{time} = 15$. This example includes an outlier (p_5) and missing points between p_6 and p_7. First, the neighborhood of p_1, denoted as $\mathbf{N}(p_1)$, is composed of three points $\{p_j \mid j \in \{1, 2, 3\}\}$ such that $d(p_j, p_1) \leq \epsilon$ and $|1-j| \leq 3$. Thus, p_1 is not a core point because the duration, $|s(\mathbf{N}(p_1))| = t_3 - t_1 = 6$ is not greater than $m_{time} = 15$. Then, the neighborhood of p_4 is composed of four points $\{p_j \mid j \in \{3, 4, 6, 7\}\}$ such that $d(p_j, p_4) \leq \epsilon$ and $|4 - j| \leq 3$. Thus, p_4 is a core point because its duration ($|t_7 - t_3| = 18$) is greater than m_{time}. p_7 is also a core point and duration-joinable to p_4 because their stay periods $[7, 25]$ and $[10, 34]$ overlap each other. As a result, D-Star can extract a cluster of non-consecutive points $C = \{p_i \mid i \in \{3, 4, 6, 7, 8, 9, 10\}\}$ in a geospatial trajectory with outliers and missing points.

As important location detection, we adopt the DBSCAN algorithm [3] which is a famous density-based clustering method. Figure 3 shows the important locations detected for user 2. The location information shown in Fig. 3, such as near a station or in a park, is useful for understanding his/her lifelog. In the rest of this paper, we use these important location indices as input features. In addition, home and work tags are delivered on behalf of GPS data while users are in their house or office. We combine them into the location index feature.

4. LAT

4.1 Our approach

4.1.1 Data annotation

The objective of the LAT subtask is to appropriately annotate fifteen concept labels to all lifelog moments having images. This task is known as multi-label classification, which is addressed by using supervised machine learning approaches in many previous works. To handle this task in the same way, the authors manually annotated multiple lifelog labels for supervised training to each moment in four days, Sept. 9, Sept. 10, Sept. 29, and Oct. 3, 2016 in the phase 1 dataset. Moreover, to decrease annotation error among authors, we additionally annotated *context labels* that are defined by authors (Table 1). Annotation criterion for these labels was less ambiguous for annotators than the fifteen lifelog labels on the annotation task because they were decided on the basis of dataset images.

4.1.2 DNN model

Our team developed a DNN model with fusion layer of the tri-modal data of image, location, and sensor, shown in Fig. 4. Each component in the DNN model encodes appropriate feature representation. The visual and location indexing processes were explained in Section 2 and 3. The sensor features consist of five values: steps, heart rate, GSR, calories, and skin temperature. These values are normalized to $\mathcal{N}(0,1)$ across all days before input. One of the features of our model is that it estimates context labels, which are defined by our team, as another task. Herewith, image encoding components can learn weight parameters from context labels in addition to lifelog labels. Each of them is encoded by a fully connected neural network to one feature representation. After that, the output layer estimates fifteen lifelog labels via fifteen sigmoid functions on the basis of the feature values of the last fully connected layer.

For DNN optimization, our team applies Adam[5] based on gradient calculated by using the cross-entropy error function. Here, the number of mini-batch size was 10, and the number of back-propagation iterations was 50.

4.1.3 Post-processing

There are topics in which the location condition was specified, such as "Commuting", "Travelling" and "In a restaurant". To tag these labels properly, we have to distinguish where the user is around that moment. However, it is difficult to determine a user's location by using only image information.

To overcome this problem, we exploited location cluster information that was obtained by D-Star and DBSCAN (see Section 3). We set the "In a restaurant" score to 0 if the moment's location cluster was not related to restaurants. Also, regarding moments of movement, if the starting moments of the movement were in the home or office cluster, the "Travelling" score was set to 0; otherwise the "Commuting" score was set to 0.

4.2 Official results

We submitted the eight runs shown in Table 2 and received officially evaluated results for them. The image, location, and sensor columns denote the presence or absence of input data. The PP column denotes whether post-processing by location condition was done or not.

[5]We set hyperparameters to $\alpha = 0.0001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$.

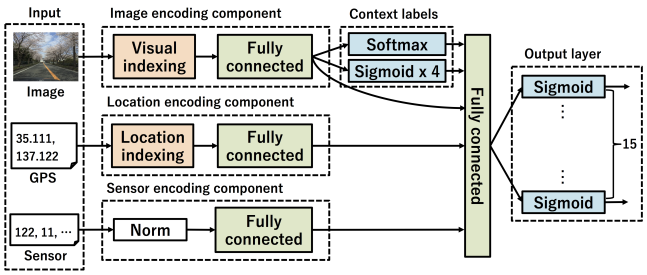

Figure 4: Our team's LAT DNN model

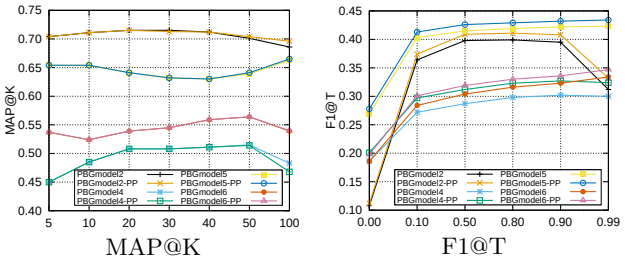

Figure 5: Evaluation values of each run

The official evaluation results are shown in Fig. 5. The left and right figures are the mean average precision (MAP) at rank K and F1 score at threshold T. For MAP@K, PBGmodel2 and PBGmodel2-PP showed the highest score among the other methods for all ranks. In comparison, the highest score for F1@T for all threshold values was achieved by PBGmodel5-PP. These results suggest that the visual and sensor features can enhance the automatic annotation performance; however, the location features reduce it.

To clarify whether topics are difficult or easy topics with our methods, we show the F1@T=0.9 score of each method for all topics in Fig. 6. The result suggests that LAT001, LAT002, LAT008, LAT012, and LAT014 are easy topics, and LAT003, LAT006, LAT007, and LAT013 are difficult topics, relatively. In particular, for LAT007, which is "Watching TV," the F1@T scores of our all methods were zero because we could not find images with LAT007 in our annotation phase. For LAT009, which is "Exercise," the evaluation scores of PBGmodel5-PP and PBGmodel5 were higher than the other methods. We think that these two methods could accurately train the context of exercise by using the sensor feature. Finally, we confirmed the effectiveness of the post-processing from the evaluation score with LAT001 (Commuting), LAT002 (Travelling), and LAT012 (In a restaurant) because the PBGmodel5-PP's evaluation scores for these topics were higher compared with those of the PBGmodel5.

5. LSAT

In this subtask, we have to retrieve a number of specific moments in a lifelogger's life. Moments are defined as semantic events or activities that happened throughout the day. The LSAT query consists of 24 topics. A half of them are tagged as "easy," and the rest are tagged as "difficult."

Table 1: Our defined context labels for solving LAT

Context label	Label type	Description
Conversation	binary	Does this image include a conversation scene?
Other people	binary	Does this image include other people?
Tourist site	binary	Does this image include a tourist site?
Eating activity	binary	Does this image include any eating related activity?
Place and Movement	categorical	Which is this image's place category home, work, commuting, staying outside of town of residence, or other?

Table 2: LAT submission runs of our team

Run ID	Image	Location	Sensor	PP
PBGmodel2	✓			
PBGmodel2-PP	✓			✓
PBGmodel4	✓	✓		
PBGmodel4-PP	✓	✓		✓
PBGmodel5	✓		✓	
PBGmodel5-PP	✓		✓	✓
PBGmodel6	✓	✓	✓	
PBGmodel6-PP	✓	✓	✓	✓

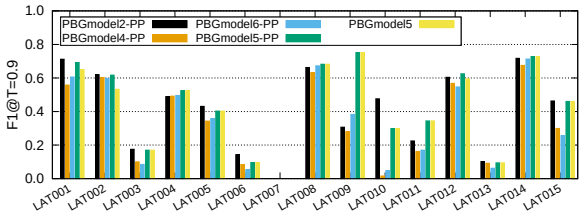

Figure 6: F1@T=0.9 scores for each topic

We processed all the query topics automatically although the task description says "the difficult topics are designed for interactive systems."

5.1 Our approach

Essentially, we adopted the usual text retrieval method for this task, where each document has a document ID, i.e., image id, and its content is a set of labels assinged to the corresponding image. There were two type of labels that we used. Labels of the first type are given in the fields of the dataset, such as official concepts, food logs, drink logs, place names, and activities. Text strings in free description fileds (food logs, drink logs, place names, and activities) are split into words, i.e., labels. We counted the document frequency for each distinct label to get the idf (inverse document frequency). Labels of the second type were assigned by our team using DNNs as described in Section 2. Stemming, case folding, and stopword removal were done on all the labels and query terms.

5.1.1 *Query processing*

We processed query topics and calculated scores largely in the usual text retrieval manner. For each query topic, we calculate a relevance score by matching query terms against labels assinged to images. We use all of the "title," "description," and "narrative" part of each query topic and assign a "weight" to each part to calculate the score. Also, for labels, we assign weights with respect to different fields and types. These weights are ajusted in order to get the best result.

5.1.2 *Scoring*

We calculated the relevance score $score(d, q)$ between each query q and each image d by using the following formula,

$$score(d, q) = \sum_{e \in E} \{w_e \cdot \sum_{c \in C} w_c \cdot rel(d_c, q_e)\} \quad (1)$$

, where E denotes the set of "title," "description," and "narrative," and C denotes the set of image features shown in Table 3. w_e is the weight parameter for controlling each query's part e, and w_c plays the role of a switching function for using each image feature c. d_c denotes the set of labels for each image feature c, and q_e denotes the set of terms for each query's part e. $rel(d_c, q_e)$ denotes the relevance value between d_c and q_e and is separately defined by image feature c. When c is `Concept` or `FDPA`, $rel(d_c, q_e)$ is calculated as follows,

$$rel(d_c, q_e) = \sum_{i \in d_c} \log_{10}(1 + \mathrm{tf}_{i,q_e}) \cdot \mathrm{idf}_i \quad (2)$$

, where tf_{i,q_e} denotes the term i frequency in query q_e, and idf_i denotes the term i inversed document (image) frequency.

When c is `ImageNet`, `Places`, `ImageNet-phone`, or `Places-phone`, $rel(d_c, q_e)$ is calculated as follows,

$$rel(d_c, q_e) = \sum_{i \in d_c} \log_{10}(1 + \mathrm{tf}_{i,q_e}) \cdot imgscore(d, i) \quad (3)$$

, where $imgscore(d, i)$ denotes the image score between image d and label i and is calculated by summing the deep neural networks' scores.

Ranking is done according to the relevance score calculated above.

5.1.3 *Temporal filter*

Some of the query topics have a rather strong time of day constraint. For example "Eating lunch" only happens near noon. Following [10], we adopt a temporal filter for such query topics. We manually add a "temporal" part to each query that indicates the time of day constraint.

After scoring, we examine whether each image satisfies the time of day constraint of the topic being processed. Images that do not satisfy this constraint are discarded from the result.

5.2 Official results

We submitted the seven runs shown in Table 4 and received official evaluated results for them. The "Official features" and "Additional features" columns denote the image features used for each method. The "Weight parameter" column represents w_e in formula (1), which is the weight of the linear combination of the title score, description score, and narrative score.

Table 3: Image features for LSAT

Feature		Description
`Concept`	(Official)	Image labels provided by organizers.
`FDPA`	(Official)	Bag of words of *food, drink, place names*, and *activities* associated with each image ID.
`ImageNet`	(Additional)	Sum of the output scores of GoogLeNet and AlexNet trained on ImageNet applied to each image.
`Places`	(Additional)	Sum of the output scores of GoogLeNet, AlexNet, VGG, and ResNet trained on Places365 applied to each image.
`ImageNet-phone`	(Additional)	`ImageNet` scores calculated for eacch phone image associated with an image ID.
`Places-phone`	(Additional)	`Places` scores calculated for each phone images associated with an image ID.

The official evaluation results of each method are shown in Table 5. PBGLSAT02, PBGLSAT07, PBGLSAT03 showed the highest precision, recall and F1 score, respectively.

Figure 7 shows the results of each topic. We overview the relationships between topics and our methods from the perspective of F1 scores. For topics such as LSAT004 (Coffee), LSAT011 (Cooking) and LSAT014 (Photo of the Sea), the method using only additional features (PBGLSAT03) showed high performance. For these topics, official concepts of related images were insufficient, so we had to extract features from images for accurate retrieval. For LSAT002 (Gardening) and LSAT006 (Graveyard), the method using `Places` features (PBGLSAT03) showed good performance. This is because it is necessary to capture a user's location to process these topics, and `Places` features include a lot of information about a user's places. The methods using "Places" features also showed good performance for topic LSAT016 (Greek Amphitheatre), although this topic was not evaluated officially because of the difficulty. For LSAT015 (Having Beers in a Bar), the features from cellphone images seemed to contribute to achieving good performance (PBGLSAT005). We consider that this is caused by the clear appearances of beers in cell-phone images. However, for some topics such as LSAT008 (Grocery Shopping), information from phone images worked as noise and degraded the performance of retrieval.

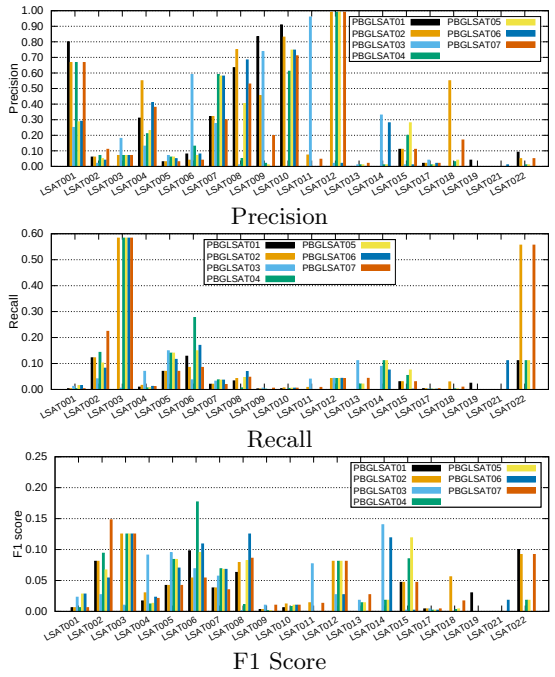

Figure 7: LSAT evaluation scores for all topics

6. LEST

The objective of the LAT subtask is to detect event segments from continual lifelog stream data. Each segment needs to consist of a start and end image-ID. We mainly prepared four types of approaches and submitted ten runs with several hyper-parameters that were decided by using supervised data prepared by the authors. In common, our approaches detect the end image-ID list in each day. Each segment is generated by a start image-ID, which is detected by detecting the end image-ID of the previous segment, and the end image-ID is then identified. Therefore, our approaches do not enumerate "untagged" segments.

6.1 Our approach

6.1.1 *Similarity based approach*

The simplest idea of event segmentation is to calculate cosine similarity between previous and next images and to enumerate image IDs with similarity under a threshold value. Two images similarities can be to used for calculation using features such as concept labels, ImageNet, and Places. As a result of our experimental evaluation, we employed cosine similarity by using GoogLeNet's ImageNet scores and identified image IDs with it under 0.04 as a segment image.

6.1.2 *D-Star-based approach*

D-star is a robust method of extracting stay regions as described in Section 3. However, it focuses only on positions and time and does not use images. We made two extensions in order to use it for images. The first extension is to use similarity-neighborhoods $\mathbf{N}_{\mathrm{sim}}(p_i)$ instead of neighborhoods $\mathbf{N}(p_i)$. Specifically, we changed the contents of the IF statement in line 5 of Algorithm 1 to $d(p_i, p_j) \leq \epsilon \wedge \mathrm{sim}(p_i, p_j) \geq \tau$. Here, τ represents the similarity threshold, and function $\mathrm{sim}(p_i, p_j)$ represents the similarity between the image of p_i and one of p_j. We defined it as

$$\mathrm{sim}(p_i, p_j) = \frac{\sum_{k \in d_C} \min(p_{i,k}, p_{j,k}) \times \mathrm{idf}_k}{\sum_{k \in d_C} \max(p_{i,k}, p_{j,k}) \times \mathrm{idf}_k}, \qquad (4)$$

with reference to [13], where $p_{i,k}$ is an image feature calculated by a deep neural network. In this paper, we used Places scores calculated by GoogLeNet with $p_{i,k}$. The other extension regards duration-joinable core points. If each similarity-neighborhood of two core points has the same point, they

Table 4: LSAT submission runs of our team

Run ID	Official features		Additional features				Weight parameters		
	Concept	FDPA	ImageNet	Places	ImageNet-phone	Places-phone	Title	Desc.	Narra.
PBGLSAT01	✓						1.00	0.50	0.25
PBGLSAT02	✓	✓					1.00	0.50	0.25
PBGLSAT03			✓	✓	✓	✓	1.00	0.50	0.25
PBGLSAT04	✓	✓		✓		✓	1.00	0.50	0.25
PBGLSAT05	✓	✓	✓	✓	✓	✓	1.00	0.50	0.25
PBGLSAT06	✓	✓	✓	✓			1.00	0.50	0.25
PBGLSAT07	✓	✓		✓			1.00	1.00	1.00

Table 5: LSAT precision, recall, and F1 score averaged over all topics. Highest score in each column is shown in bold.

Run ID	Precision	Recall	F1 score
PBGLSAT01	0.212	0.028	0.027
PBGLSAT02	**0.278**	0.081	0.038
PBGLSAT03	0.232	0.040	**0.047**
PBGLSAT04	0.187	0.077	0.040
PBGLSAT05	0.193	0.072	0.042
PBGLSAT06	0.215	0.068	0.043
PBGLSAT07	0.222	**0.087**	0.041

are duration-joinable. We set each parameter to $q = 5, \epsilon = 40\,\mathrm{m}, m_{time} = 3\,\mathrm{min}, T_{stay} = 5\,\mathrm{min}, \tau = 0.4$ for user 1 and $q = 5, \epsilon = 120\,\mathrm{m}, m_{time} = 3\,\mathrm{min}, T_{stay} = 5\,\mathrm{min}, \tau = 0.3$ for user 2.

6.1.3 *T-test approach*

The problem with similarity-based approaches is that noisy images can be wrongly detected as segments because the similarity between a previous image and next noisy image results in a low value. To overcome this problem, we propose a segmentation method composed from two steps; first, we reduce dimensions with images by latent Dirichlet allocation (LDA) to several latent topics [1] because the number of features dimensions is too high. Second, we apply Welch's t-test to detect unnatural image groups against current image segments by using latent topic probability in a sliding window. Concretely, when the image index currently being focused on is i $(0 \leq i < L)$ and the sliding window size is w (> 0), we extract two segments $\{i, i-1, \ldots, i-w\}$ and $\{i, i+1, \ldots, i+w\}$ and calculate the Welch's t-test value between the two segments for each latent topic number. When a summation value among all topics is over the threshold value, we detect the image ID i as the end of the segment. From our experimental evaluation, we set the number of topics in LDA to 10 and extracted 50 image IDs for each day in descending order of the summation value of the above process.

6.1.4 *Gated CNN approach*

For using various features together and checking the benefit of each feature, we used a neural network with gated convolutional neural network (CNN) module [2]. Figure 8 shows the network structure of our model. The gated CNN module can obtain features from several pieces of time-step data, and we can obtain the features of long-term time series data by using the module iteratively. We estimated whether an image is the end of a segment or not by inputting a total of thirteen images centered a target image.

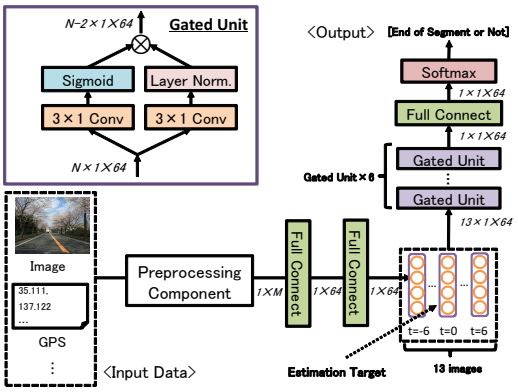

Figure 8: Gated CNN model for LEST. M indicates number of features of each image. t indicates time index of image.

As preprocessing, we obtained features from images and GPS trajectories by using the methods shown in Section 2, Section 3, and Section 6.1.3. We set the similarity feature of each image as a cosine similarity from a previous image calculated from visual index vectors obtained as described in Section 2 or LDA topics vectors obtained as described in Section 6.1.3. After that, these features are encoded into one vector by using two fully connected layers. The encoded vectors are forwarded through a gated unit iteratively, and finally, the network outputs whether the estimation target image is the end of a segment or not after a fully connected layer and a softmax layer. For optimization, our team applied Adam[6] based on gradient calculated by using cross-entropy error function. Our supervised data was imbalanced because there were much fewer ends of segment images than non-end images, thus, we picked three non-end images for one end image as learning data. The number of mini-batch sizes was 10, and the number of back-propagation iterations was 200.

6.2 Official results

Table 6 shows the list of runs our team submitted for the LEST task. We submitted six runs for the gated CNN approach with difference features. PBG-LEST01 and 04–08 used the same model for both users, and the other runs used different models. We set the parameters of each model, so that its F1 score for supervised data was prepared by the authors.

[6] We set hyperparameters to $\alpha = 0.00001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$.

Run ID	Model	Feature						Same model for both users
		Image similarity based on visual index	Image LDA topics	Image similarity based on LDA topics	GPS	Location index	Activity	
PBG-LEST01	Image similarity	✓						✓
PBG-LEST02	Sim D-Star	✓			✓			
PBG-LEST03	T-test		✓					
PBG-LEST04	T-test + D-Star		✓		✓			
PBG-LEST05	Gated CNN	✓						✓
PBG-LEST06	Gated CNN	✓					✓	✓
PBG-LEST07	Gated CNN	✓			✓			✓
PBG-LEST08	Gated CNN	✓		✓	✓		✓	✓
PBG-LEST09	Gated CNN	✓		✓	✓		✓	
PBG-LEST10	Gated CNN	✓	✓	✓	✓	✓	✓	

Table 6: LEST submission runs.

Table 7: LEST precision, recall, and F1 score averaged over all days. Highest score in each column is shown in bold.

Run ID	Precision	Recall	F1 score
PBG-LEST01	**0.901**	0.352	0.494
PBG-LEST02	0.559	**0.698**	**0.579**
PBG-LEST03	0.768	0.453	0.550
PBG-LEST04	0.762	0.485	0.573
PBG-LEST05	0.848	0.421	0.547
PBG-LEST06	0.837	0.421	0.545
PBG-LEST07	0.855	0.406	0.535
PBG-LEST08	0.860	0.407	0.539
PBG-LEST09	0.846	0.436	0.561
PBG-LEST10	0.790	0.455	0.551

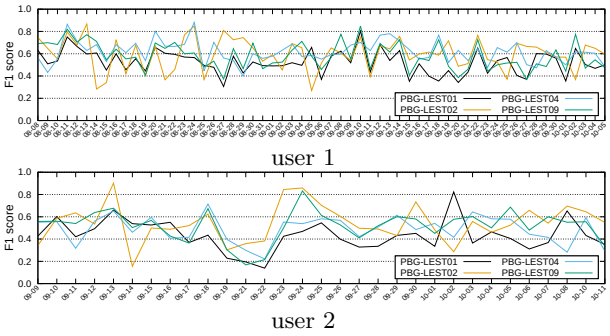

Figure 9: LEST F1 score of four runs (Simplest Method:01, D-Star based:02, T-test based:04, gated-cnn: 09) in all days

Table 7 shows the evaluation of each run. Fig. 9 shows the score for each day of four methods. PBG-LEST01, which used the simplest image similarity based approach, achieved the highest precision score; however, its recall score was quite worse than other runs. This result indicates that two images having low similarity became almost a segment boundary; however, there is often a boundary without a decrease in similarity. PBG-LEST02, which used the D-Star based approach, achieved the highest recall score and F1 score, and its F1 score was the highest among all the runs of this subtask. PBG-LEST02 could detect a segment boundary accurately by considering the movement states of a user with his/her GPS data. For example, Fig. 9 shows that the F1 score of PBG-LEST02 for the 09–13 of user 2 was quite higher than the other runs. One that day, user 2 went shopping in a shopping mall and did some work at his house while moving. During these segments, user 2 kept on moving; thus, segmentation based on image similarity was difficult because image similarity was always low. In comparison, our D-Star based approach can consider a user's movement state with GPS data. As a result, the score for LEST02 for that day was high. Regarding the T-test approach, the run with GPS data (PBG-LEST04) was more accurate than that without GPS (PBG-LEST03). This result also shows the importance of considering location information. The gated CNN approach with the LDA topics feature (PBG-LEST10) was less accurate than without that feature (PBG-LEST09). This may be because the number of dimensions of the LDA topics feature was too large for supervised data; thus, our model was overfitting.

7. CONCLUSIONS

To solve the Lifelog Annotation (LAT), Lifelog Semantic Access (LSAT), Lifelog Event Segmentation (LEST) subtasks, we proposed effective models based on visual and location indexing methods. From official results and our qualitative evaluation, we demonstrated their outstanding performance and clarified the effectiveness and limitation of each data analyzing function and lifelog feature.

As future work, we will focus on classifying segments into semantic categories by using a small set of labeled data. Moreover, we will consider a novel DNN framework for representation learning from multi-modal and time-series data.

8. REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.

[2] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier. Language modeling with gated convolutional networks. *arXiv preprint arXiv:1612.08083*, 2016.

[3] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.

[4] C. Gurrin, H. Joho, F. Hopfgartner, L. Zhou, D. T. D. Nguyen, R. Gupta, and R. Albatal. Overview of the NTCIR-13 lifelog-2 task. In *The NTCIR-13 Conference*, Tokyo, Japan, 2017.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern*

Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 770–778, 2016.

[6] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA, 2014. ACM.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, NIPS'12, pages 1097–1105. Curran Associates Inc., USA, 2012.

[8] K. Nishida, H. Toda, and Y. Koike. Extracting arbitrary-shaped stay regions from geospatial trajectories with outliers and missing points. In *ACM SIGSPATIAL International Workshop on Computational Transportation Science (IWCTS)*, pages 1–6, 2015.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[10] B. Safadi, P. Mulhem, G. Quénot, and J. Chevallet. LIG-MRIM at NTCIR-12 lifelog semantic access task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies*, NTCIR-12, pages 361–365, 2016.

[11] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, June 2015.

[13] Y. Takimoto, K. Sugiura, and Y. Ishikawa. Extraction of frequent patterns based on users' interests from semantic trajectories with photographs. In *Proceedings of the 21st International Database Engineering & Applications Symposium*, pages 219–227. ACM, 2017.

[14] Y. Ye, Y. Zheng, Y. Chen, J. Feng, and X. Xie. Mining individual life pattern based on location history. In *2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, pages 1–10. IEEE, 2009.

[15] Y. Zheng and X. Zhou. *Computing with spatial trajectories*. Springer Science & Business Media, 2011.

[16] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.