BIDAL@imageCLEFlifelog2019: The Role of Content and Context of Daily Activities in Insights from Lifelogs

Minh-Son Dao[1], Anh-Khoa Vo[2], Trong-Dat Phan[2], and Koji Zettsu[1]

[1] Big Data Analytics Laboratory
National Institute of Information and Communications Technology, Japan
{dao,zettsu}@nict.go.jp
[2] Faculty of Information Technology
University of Science, VNU-HCMC, Vietnam
{1512262,1512102}@student.hcmus.edu.vn

Abstract. imageCLEFlifelog2019 introduces two exciting challenges of getting insights from lifelogs. Both challenges aim to have a memory assistant that can accurately bring a memory back to a human when necessary. In this paper, two new methods to tackle these challenges by leveraging the content and context of daily activities are introduced. Two backbone hypotheses are built based on observations: (1) under the same context (e.g., a particular activity), one image should have at least one associative image (e.g., same content, same concepts) taken from different moments. Thus, a given set of images can be rearranged chronologically by ordering their associative images whose orders are known precisely, and (2) a sequence of images taken during a specific period can share the same context and content. Thus, if a set of images can be clustered into sequential atomic clusters, given an image, it is possible to automatically find all images sharing the same content and context by first finding the atomic cluster sharing the same content, then watershed reward and forward to find other clusters sharing the same context. The proposed methods are evaluated on the imageCLEFlifelog 2019 dataset and compared to participants joined this event. The experimental results confirm the high productivity of the proposed method in both stable and accuracy aspects.

Keywords: lifelog · content and context · watershed · image retrieval · image rearrange

1 Introduction

Recently, research communities start getting used with new terminologies "lifelogging" and "lifelog". The former represents the activity of continuously recording people everyday experiences. The latter implies the dataset contained data

generated by lifelogging. The small sizes and affordable prices of wearable sensors, high bandwidth of the Internet, and flexible cloud storages encourage people to lifelogging more frequent than before. That leads to the fact that people can have more opportunities to understand their lives thoroughly due to daily recording data whose content conceal both cognitive and physiological information. One of the most exciting topics when trying to get insights from lifelogs is to understand human activities from the first-person perspective[6][7]. Another interesting topic is to augment human memory towards improving human capacity to remember[10]. The former aims to understand how people act daily towards having effective and efficient support to improve the qualification of living both in social and physical activities. The latter tries to create a memory assistant that can accurately and quickly bring a memory back to a human when necessary.

In order to encourage people to pay more attention to the topics above, several events have been organized[8][9][2][4]. These events offer a large annotated lifelog collected from various sensors such as physiology (e.g., heartbeat, step counts), images (e.g., lifelog camera), location (e.g., GPS), users tags, smartphone logs, and computer logs. A series of tasks introduced throughout these events started attracting people leading to an increase in the number of participants. Unfortunately, the results of proposed solutions from participants are far from expectation. It means that lifelogging still a mystical land that needs to be discovered more both in what kind of insights people can extract from lifelogs and how the accuracy of these insights are.

Along this direction, imageCLEFlifelog2019 - a session of imageCLEF2019[11] - is organized with two tasks[3]: (1) solve my puzzle: the new task introduced this time, and (2) lifelog moment retrieval: the old one with some modification to make it more difficult and enjoyable than before. In this paper, two new methods are introduced to tackle two tasks mentioned. The basement of the proposed methods is built by utilizing the association of contents and contexts of daily activities. Two backbone hypotheses are built based on this basement:

1. Under the same context (e.g., a particular activity), one image should have at least one associative image (e.g., same content, same concepts) taken from different moments. Thus, a given set of images can be rearranged chronologically by ordering their associative images whose orders are known precisely, and
2. A sequence of images taken during a specific period can share the same context and content. Thus, if a set of images can be clustered into sequential atomic clusters, given an image, it is possible to automatically find all images sharing the same content and context by first finding the atomic cluster sharing the same content, then watershed reward and forward to find other clusters sharing the same context.

The paper is organized as follows: Section 2 and 3 introduce the solutions for Task1 and Task 2, respectively. Section 4 describes the experimental results gotten by applying two proposed methods as well as related discussions. The last Section gives conclusions and future works.

2 Task 1: Solve my puzzle

In this section, we introduce the proposed method, as well as its detail explanation, algorithms, and examples related to the Task 1.

Task 1: *solve my life puzzle* is stated as: *Given a set of lifelogging images with associated metadata such as biometrics and location, but no timestamps, these images are required to be rearranged in chronological order and predict the correct day (e.g., Monday or Sunday) and part of the day (morning, afternoon, or evening)*[3].

2.1 From Chaos to Order

We build our solution based on one of the characteristics of lifelogs: activities of daily living (ADLs)[5]. Some common contents and contexts people live in every day can be utilized to associate unordered time images to ordered time images. For example, one always prepares and has breakfast in a kitchen every morning from 6 am to 8 am, except for a weekend. Hence, all record r_i^d recorded from 6 am to 8 am in the kitchen are shared the same context (i.e., in a kitchen, chronological order of sequential concepts) and content (e.g., objects in a kitchen, foods). If we can associate one-by-one each image of a set of unordered time images r_i^q to a subset of ordered time images r_i^d, r_i^q can totally be ordered by the order of r_i^d. This above observation brings the following useful hints:

- If one record=(image, metadata) r^q captured in the scope of ADLs, there probably is a record r^d sharing the same content and context.
- If we can find all associative pairs (r_i^d, r_i^q), we can rearrange r_i^q by rearranging r_i^d utilizing metadata of r_i^d, especially timestamps and locations.

We call r^q and r^d a *query record/image* and *associative record/image* (of the query record/image), respectively, if $similary(r^d, r^q) > \theta$ (the predefined threshold). We call a pair of (r^d, r^q) an *associative pair* if it satisfies the condition $similary(r^d, r^q) > \theta$. Hence, we propose a solution as follows: Given sets of unordered images $Q = \{r_i^q\}$ and ordered images $D = \{r_i^d\}$, we

1. Find all associative pairs (r^d, r^q) by using the similarity function (described in subsection 3.2). The output of this stage is a set of associative pairs ordering by timestamps of D, call P. Next, P is refined to generate P_{TOU} to guarantee that there is only one associate pair remained with the highest similarity score among consecutive associative pairs of the same query image. Algorithm 1 models this task. Fig. 1, the first and second rows, illustrates how this task works.
2. Extract all patterns *pat* of associative pair to create P_{UTOU}. The pattern is defined as the set of associative pairs so that the query images set is precisely the same as the images of Q. Algorithm 2 models this task. Figure 1, the third row 27 denotes the process of extracting patterns.

3. Find the best pattern P_{BEST} that has the highest average similarity score comparing to the rest. Algorithm 3 models this task. Fig. 1, the third row denotes the process of selecting the best pattern.
4. The unordered images Q is ordered by order of associative images of P_{BEST}. Fig. 1, the red rectangle at the left-bottom corner illustrates this task.

2.2 Similarity Functions

The similarity function to measure the similarity between a query record and its potentially associative record has two different versions built as follows:

1. **Color histogram vector and Chi-square**: The method introduced by Adrian Rosebrock[3] is utilized to build the histogram vector. As mentioned above, the environment of images taken by lifelogging can be repeated daily, both indoor and outdoor. Hence, the color information probably is the primary cue to find similar images captured under the same context.
2. **Category vector and FAISS**: The category vector is concerned as a deep learning feature to overcome problems that handcraft features cannot do. The ResNet18 of the pre-trained model PlaceCNN[4] is utilized to create the category vector. The 512-dimension vector extracted from the "avgpool" layer of the ResNet18 is used. The reason such a vector is used is to search images sharing the same context (spatial dimension). The FAISS (Facebook AI Similarity Search)[5] is utilized to push the speed of searching for similar images due to its high productivity of similarity searching and clustering dense vectors.

2.3 Parameters Definitions

Let $Q = \{q[k]\}_{k=1..\|Q\|}$ denote the set of query images, where $\|Q\|$ is the total number of images in the query set.

Let $D = \{d[l]\}_{l=1..\|D\|}$ denote the set of images contained in the given data set, where $\|D\|$ is the total number of images in the data set.

Let $P = \{p(iD, iQ)[m]\}$ denote the set of associative indices that point to related images contained in D and Q, respectively. In other words, using $p[m].iD$ and $p[m].iQ$ we can access images $d[p[m].iD] \in D$ and $q[p[m].iQ] \in Q$, respectively.

Let P_{TOU} and P_{UTOU} denote the sets of time-ordered-associative (TOU) images and unique-time-ordered-associative (UTOU) images, respectively.

Let $P_{BEST} = \{p(iD, iQ)[k]\}$ denote the best subset of P_{UTOU} satisfy

- $\forall m \neq n : p[m].iQ \neq p[n].iQ$ AND $\|P_{BEST}\| == \|Q\|$ (i.e., unique)

[3] https://www.pyimagesearch.com/2014/12/01/complete-guide-building-image-search-engine-python-opencv

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

[5] https://code.fb.com/data-infrastructure/faiss-a-library-for-efficient-similarity-search

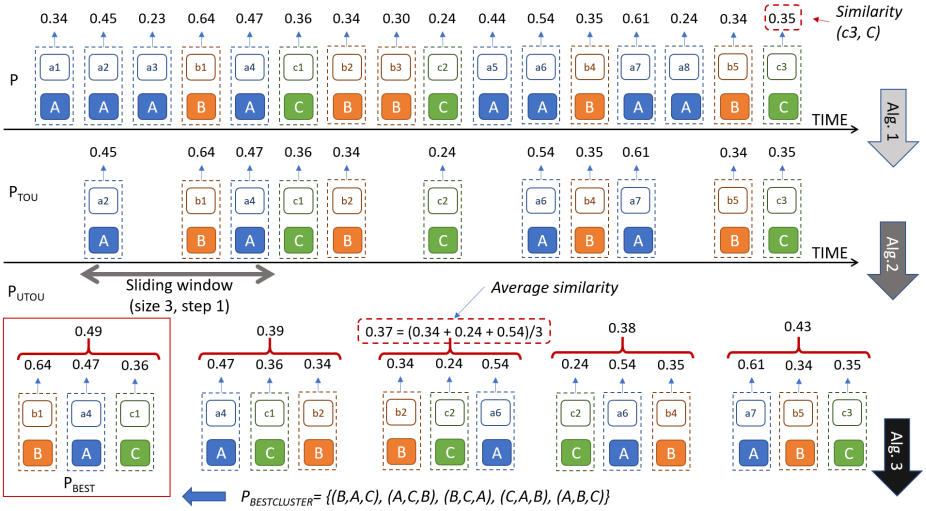

Fig. 1. Task 1: An example of finding P_{TOU}, P_{UTOU}, and P_{BEST} where $Q = \{A, B, C\}$, $\{a_i, b_j, c_k\} \in D$ are ordered by time, $\theta = 0.24$

- $\forall m : time(p[m].iD) < time(p[m+1].iD)$ (i.e., time-ordered)
- $\frac{1}{k}\sum_{i=1}^{k}(similarity(d[p[i].iD], q[p[i].iQ])) \Rightarrow MAX$ (i.e., associative)

Let θ, α, and β denote the similarity threshold, the number of days in the data set, and the number of clusters within one day (i.e. parts of the day), respectively.

3 Task 2: Lifelog Moment Retrieval

In this section, we introduce the proposed method, as well as its detail explanation, algorithms, and examples related to Task 2.

Task 2: *Lifelog moment retrieval* is stated as[3]: *Retrieve a number of specific predefined activities in a lifelogger's life.*

3.1 The Interactive-Watershed-based for Lifelog Moment Retrieval

The proposed method is built based on the following observation: A sequence of images taken during a specific period can share the same context and content. Thus, if a set of images can be clustered into sequential atomic clusters, given an image, we can automatically find all images sharing the same content and context by first finding the atomic cluster sharing the same content, then watershed reward and forward to find other clusters sharing the same context. The terminology atomic cluster is understood that all images inside should share the similarity higher than the predefined threshold and must share the same context (e.g., location, time).

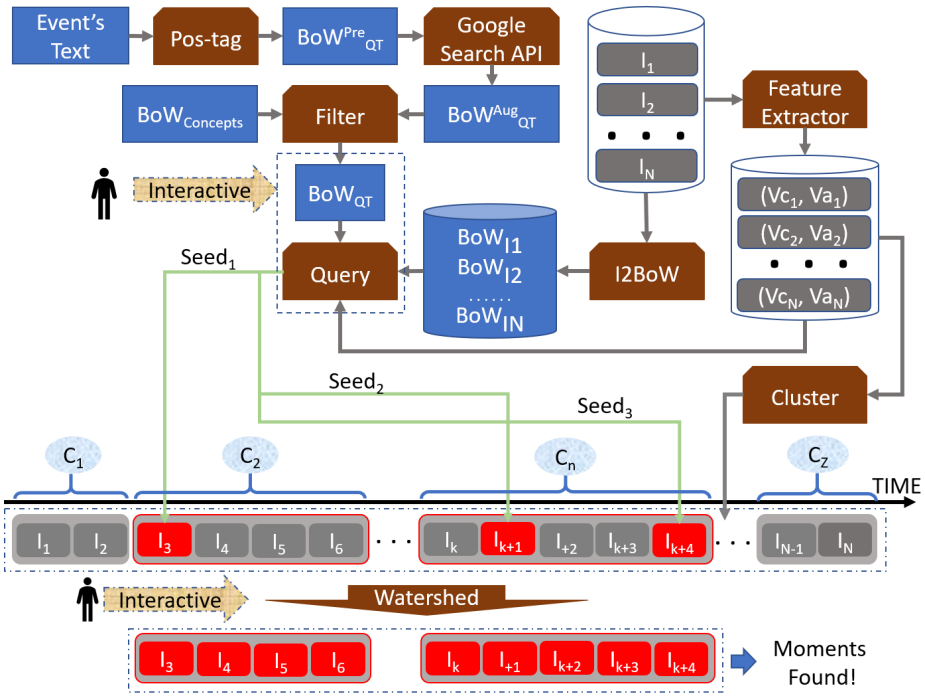

Fig. 2. An Overview of the Proposed Method for Task 2: Interactive watershed-based lifelog moment retrieval

Based on the above discussion, the Algorithm 1 is built to find a Lifelog moment from a dataset defined be a query events text. Following is the description to explain the Algorithm 1.

- **Stage 1** (Offline): This stage aims to cluster a dataset into the atomic clusters. The category vector Vc and attribute vector Va extracted from images are utilized to build the similarity function. Besides, each image is analyzed by using *I2BoW* function to extract its BoW contains concepts and attributes reserved for later processing.
- **Stage 2** (Online): This stage targets to find all clusters satisfied a given query. Since the given query is described by text, a text-based query method must be used for finding related images. Thus, we create *Pos_tag*, *Google-SearchAPI*, and *Filter* functions to find the best BoWs that represent the taxonomy of the querys context and content. In order to prune the output of these functions, we utilize the *Interactive* function to select the best taxonomy. First, images queried by using BoW (i.e., seeds) are utilized for finding all clusters, namely LMRT1, that contain these images. These clusters are then hidden for the next step. Next, these seeds are used to query on the rest clusters (i.e., unhidden clusters) to find the second set of seeds. These second set of seeds are used to find all clusters, namely LRMT2, that contain

these seeds. The final output is the union of C1 and C2. At this step, we might apply *Interactive* function to refine the output (e.g., select clusters manually). Consequently, all lifelog moment satisfied the query are found.

3.2 Functions

In this subsection, the significant functions utilized in the proposed method are introduced, as follows:

- **Pos_tag**: processes a sequence of words tokenized from a given set of sentences, and attaches a part of speech tag (e.g., noun, verb, adjective, adverb) to each word. The library NLTK[6] is utilized to build this function.
- **featureExtractor**: analyzes an image and return a pair of vectors v_c (category vector, 512 dimension) and v_a (attribute vector, 102 dimension). The former is extracted from the "avgpool" layer of the RestNet18 of the pretrained model PlaceCNN [13]. The latter is calculated by using the equation $v_a = W_a^T v_c$, introduced in [12], where W_a is the weight of Learnable Transformation Matrix.
- **similarity**: measures the similarity between two vectors using the cosine similarity.
- **I2BoW**: converts an image into a bag of words using the method introduced in [1]. The detector developed by the authors return concepts, attribute, and relation vocab. Nevertheless, only a pair of attribute and concept is used for building I2BoW.
- **GoogleSearchAPI**: enriches a given set of words by using Google Search API [7]. The output of this function is the set of words that could be probably similar to the queried words under certain concepts.
- **Interactive**: allows users to interfere with refining the results generated by related functions.
- **Query**: finds all items of the searching dataset that are similar to queried items. This function can adjust its similarity function depending on the type of input data.
- **cluster**: clusters images into sequential clusters so that all images of one cluster must share the highest similarity comparing to its neighbors. All images are sorted by time before being clustered. Algorithm 2 describes this function in detail. Figure 3.1 illustrates how this function works.

3.3 Parameter Definitions

Let $I = \{I_i\}_{i=1..N}$ denote the set of given images (e.g., dataset).

Let $F = \{(Vc_i, Va_i)\}_{i=1..N}$ denote the set of feature vectors extracted from I.

Let $C = \{C_k\}$ denote a set of atomic clusters.

[6] https://www.nltk.org/book/ch05.html
[7] https://github.com/abenassi/Google-Search-API

Let $\{BoW_{I_i}\}_{i=1..N}$ denote the set of BoWs; each of them is a BoW built by using the $I2BoW$ function.

Let $\{BoW_{QT}\}, \{BoW_{QT}^{Noun}\}, \left\{BoW_{QT}^{Aug}\right\}$ denote the Bag of Words extracted from the query, the NOUN part of BoW_{QT}, and the augmented part of BoW_{QT}, respectively.

Let $Seed_j^i$ and $LMRT^k$ denote a set of seeds and lifelog moments, respectively.

4 Experimental Results

In this section, datasets and evaluation metrics used to evaluate the proposed solution are introduced. Besides, the comparison of our results with others from different participants is also discussed.

4.1 Datasets and Evaluation Metrics

We use the dataset released by the imageCLEFlifelog 2019 challenge[3]. The challenge introduces an entirely new rich multimodal dataset which consists of 29 days of data from one lifeloggers. The dataset contains images (1,500-2,500 per day from wearable cameras), visual concepts (automatically extracted visual concepts with varying rates of accuracy), semantic content (semantic locations, semantic activities) based on sensor readings (via the Moves App) on mobile devices, biometrics information (heart rate, galvanic skin response, calorie burn, steps, continual blood glucose, etc.), music listening history, computer usage (frequency of typed words via the keyboard and information consumed on the computer via ASR of on-screen activity on a per-minute basis).

Generally, the organizers of the challenge do not release the ground truth for the test set. Instead, participants must send their arrangement to the organizers and get back their evaluation.

Task 1: Solve my puzzle Two training and testing query sets are given. Each of them has a total of 10 sets of images. Each set has 25 unordered images need to be rearranged. The training set has its ground truth to let participants can evaluate their solution.

For evaluating, the Kendall rank correlation coefficient is utilized to evaluate the similarity between the arrangement and the ground truth. Then, the mean of the accuracy of the prediction of which part of the day the image belongs to and the Kendall's Tau coefficient is calculated to have the final score.

Task 2: Lifelog Moment Retrieval The training and testing set stored with the JSON format, have ten queries whose titles are listed in Tables 3 and 4, respectively. Besides, more descriptions and constraints are also listed to guide participants on how to understand queries precisely.

The evaluation metrics are defined by imageCLEFlifelog 2019 as follows:

– Cluster Recall at X (CR@X): a metric that assesses how many different clusters from the ground truth are represented among the top X results,
– Precision at X (P@X): measures the number of relevant photos among the top X results,
– F1-measure at X (F1@X): the harmonic mean of the previous two.

X is chosen as 10 for evaluation and comparison.

4.2 Evaluation and Comparison

Task 1: Solve my puzzle Although eleven runs were submitted to evaluate, only the best five runs are introduced and discussed in this paper: (1) only color histogram vector, (2) color histogram vector and the constraint of timezone (e.g., only places in Dublin, Ireland) (3) object search (i.e., using visual concepts of metadata), and category vector plus the constraint of timezone, (4) timezone is strictly narrowed into only working days and inside Dublin, Ireland, and (5) only category vector, and the constraint of timezone (i.e., only places in Dublin, Ireland).

The reason we reluctantly integrate places information to the similarity function due to the uncertainty of metadata given by the organizer. Records that have location data (e.g., GPS, place names) occupy only 14.90%. Besides, most of the location data give wrong information that can lead to similarity scores coverage in a wrong optimal peak.

As described in Table 1, the similarity function using only color histogram gave the worst results compared to those who use the category vector. Since users are living in Dublin, Ireland, the same daily activities should share the same context and content. Nevertheless, if users go abroad or other cities, this context and content will be changed totally. Hence, the timezone could be seen as the majority factor that can improve the final accuracy since it constraints the context of searching scope. As mentioned above, the metadata does not usually contain useful data that can be leveraged the accuracy of searching. Thus, when utilizing object concepts as a factor for measuring the similarity, it can pull down the accuracy comparing to not using them at all. Distinguish between working day and weekend activities depending totally on types of activities. In this case, the number of activities that happen every day is more significant than those that happen only on working days. That explains the higher accuracy of ignoring the working day/weekend factor. In general, the similarity funtion integrated timezone (i.e., daily activities context) and category vector (i.e., rich content) gives the best accuracy compared to others. It should be noted that the training query sets are built by picking up exactly images from dataset while the testing query sets not appear in the dataset at all. That explains why the accuracy of the proposed method is perfect when running on training query sets.

Four participants submitted their outputs for evaluating (1) DAMILAB, (2) HCMUS (Vietnam National University in HCM city), (3) BIDAL (ourselves), and (4) DCU (Dublin City University, Ireland). Table 2 denotes the difference

Table 1. Task 1: An Evaluation of Training and Testing Sets ($\beta \Leftarrow 4, \theta \Leftarrow 0.24$)

Run	Parameters	Primary Score	
		Training	Testing
1	color_hist	1.000	0.208
2	time_zone+color_hist	1.000	0.248
3	time_zone+object_serach+category_vec	1.000	0.294
4	time_zone+working_day+category_vec	1.000	0.335
5	time_zone+category_vec	1.000	0.372

Table 2. Task 1: A Comparison with Other Methods

Query ID	Primary score			
	BIDAL	DAMILAB	DCU	HCMUS
1	0.490	0.220	0.377	**0.673**
2	0.260	0.200	0.240	**0.380**
3	0.310	0.153	0.300	**0.530**
4	0.230	0.220	0.233	**0.453**
5	0.453	0.200	0.240	**0.617**
6	0.447	0.320	0.240	**0.817**
7	0.437	0.293	0.360	**0.877**
8	0.300	0.200	0.300	**0.883**
9	**0.447**	0.300	0.270	0.280
10	**0.347**	0.357	0.120	0.020
Average	0.372	0.246	0.268	**0.553**

in primary scores between results generated by methods proposed by these participants and the proposed method. Although the final score of the proposed method is lower than of HCMUS, the proposed method might be more stable than others. In other words, the variance of accuracy scores when rearranging ten different queries of the proposed method is less than others. Hence, the proposed method can cope with underfitting, overfitting and bias problems.

Task 2: Lifelog Moment Retrieval Although three runs were submitted to evaluate, two best runs are chosen to introduce and discuss in this paper: (1) run 1: *interactive mode*. In this run, interactive functions are activated to let users interfere and manually get rid of those images that do not relevant to a query, (2) run 2: *automatic mode*. In this run, a program runs without any interfere from users.

Table 3 and 4 show the results running on the training and testing sets, respectively. Opposite to the results of Task 1, the results of Task 2 do not have much difference between training and testing stages. That could lead to the conclusion that the proposed method probably is stable and robust enough to cope with different types of queries.

Table 3. Task 2: Evaluation of all Runs on the Training Set

Event	Run 1			Run 2		
	P@10	**CR@10**	**F1@10**	**P@10**	**CR@10**	**F1@10**
01. Ice cream by the Sea	1.00	0.75	**0.86**	1.00	0.50	0.67
02. Having food in a restaurant	0.80	0.44	**0.57**	0.50	0.15	0.23
03. Watching videos	0.80	0.19	**0.31**	0.80	0.19	**0.31**
04. Photograph of a Bridge	1.00	0.02	0.03	1.00	0.02	**0.04**
05. Grocery shopping	0.30	0.56	**0.39**	0.40	0.22	0.28
06. Playing a Guitar	0.33	0.50	0.40	0.33	1.00	**0.50**
07. Cooking	1.00	0.18	**0.30**	1.00	0.15	0.26
08. Car Sales Showroom	0.50	0.86	0.63	0.50	1.00	**0.67**
09. Public transportation	0.60	0.07	**0.13**	0.60	0.07	0.12
10. Paper or book reviewing	0.29	0.42	**0.34**	0.43	0.17	0.24
Average Score	**0.66**	**0.40**	**0.40**	**0.66**	**0.35**	**0.33**

In all cases, the P@10 results are very high. It proves that the approach used for querying seeds of the proposed method is useful and precise. Moreover, the watershed-based stage after finding seeds can help not only to decrease the complexity of querying related images but also to increase the accuracy of event boundaries. Unfortunately, the CR@10 results are less accuracy comparing to P@10. The reason could come from merging clusters. Currently, clusters gained after running watershed are not merged and rearranged. That could lead to low accuracy when evaluating CR@X. This issue is investigated thoroughly in the future.

Nevertheless, misunderstanding context and content of queries sometimes lead to the worst results. For example, both runs failed in query 2 "driving home" and query 9 "wearing a red plaid shirt." The former was understood as "driving from office to home regardless of how many times stop at in-middle places," and the latter was distracted by synonym words of "plaid shirt" when leveraging GoogeSearchAPI to augmented the BoW. The first case should be understood that "driving home from the last stop before home," and the second case should focus on only "plaid shirt" not "sweater" nor "fannel." After fixing these mistakes, both runs have higher scores on query 2 and query 9, as described in Table 4. The second rows of event 2, 9, and the average score show the results after correcting the mentioned misunderstanding. Hence, building a flexible mechanism to automatically build a useful taxonomy from a given query to avoid these mistakes is built in the future.

Nine teams participated to Task 2 included (1) HCMUS, (2) ZJUTCVR, (3) BIDAL (ourselves), (4) DCU, (5) ATS, (6) REGIMLAB, (7) TUCMI, (8) UAPT, and (9) UPB. The detail information of these teams could be referred to in [3]. Table 5 denotes the comparison among these teams. We are ranked in the third position. In general, the proposed method can find all events with acceptance accuracy (i.e., no event with zero F1@10 scores comparing to others

Table 4. Task 2: Evaluation of all Runs on the Test set

Event	**Run 1**			**Run 2**		
	P@10	**CR@10**	**F1@10**	**P@10**	**CR@10**	**F1@10**
01. In a Toyshop	1.00	0.50	**0.67**	1.00	0.50	**0.67**
02. Driving home	0.00	0.00	0.00	0.00	0.00	0.00
	0.70	0.05	**0.09**	0.70	0.05	**0.09**
03. Seeking Food in a Fridge	1.00	0.22	**0.36**	0.60	0.17	0.26
04. Watching Football	1.00	0.25	**0.40**	0.50	0.25	0.33
05. Coffee time	1.00	0.11	0.20	1.00	0.22	**0.36**
06. Having breakfast at home	0.60	0.11	**0.19**	0.30	0.11	0.16
07. Having coffee with two person	0.90	1.00	**0.95**	0.90	1.00	**0.95**
08. Using smartphone outside	0.70	0.33	**0.45**	0.30	0.33	0.32
09. Wearing a red plaid shirt	0.00	0.00	0.00	0.00	0.00	0.00
	0.90	0.14	**0.25**	0.50	0.14	0.22
10. Having a meeting in China	0.70	0.33	**0.45**	0.70	0.33	**0.45**
Average Score	0.69	0.29	0.37	0.53	0.29	0.35
	0.85	**0.31**	**0.40**	**0.65**	**0.31**	**0.38**

those have at least one event with zero F1@10 scores). That confirms again the stability and anti-bias of the proposed method.

5 Conclusions

We introduce two methods for tackling two tasks challenged by imageCLEFlifelog 2019: (1) solve my puzzle, and (2) lifelog moment retrieval. Although each method is built differently, the backbone of these methods is the same: considering the association of content and context of daily activities. The first method is built for augmenting human memory when the chaos of memory snapshots must be rearranged chronologically to give a whole picture of a users life moment. The second method is constructed to visualize peoples memories from their explanation: start from their pinpoints of memory and watershed to get all image clusters around those pinpoints. The proposed method is thoroughly evaluated by the benchmark dataset provided by the imageCLEFlifelog 2019, and compared with other solutions coming from different teams. The final results show that the proposed method is developed in the right direction even though it needs more improvement to reach the expected targets. Many issues are raised during the experimental results and need to be investigated further for better results. In general, two issues should be investigated more in the future: (1) Similarity functions and watershed boundaries, and (2) A taxonomy generated from queries.

References

1. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question

Table 5. Task 2: Comparison to Others (Avg: descending order from left to right)

Event ID	F1@10								
	HCMUS	**ZJUTCVR**	**BIDAL**	**DCU**	**ATS**	**REGIMLAB**	**TUC_MI**	**UAPT**	**UPB**
01. In a Toyshop	**1.00**	0.95	0.67	0.57	0.46	0.44	0.46	0.00	0.29
02. Driving home	**0.42**	0.17	0.09	0.09	0.09	0.06	0.13	0.00	0.08
03. Seeking Food in a Fridge	0.36	0.40	0.36	0.29	0.36	**0.43**	0.00	0.07	0.07
04. Watching Football	**0.86**	0.37	0.40	0.67	0.00	0.40	0.00	0.00	0.00
05. Coffee time	**0.68**	0.47	0.20	0.36	0.36	0.54	0.00	0.11	0.34
06. Having breakfast at home	0.00	**0.35**	0.19	0.17	0.26	0.00	0.18	0.14	0.00
07. Having coffee with two person	0.89	**1.00**	0.95	0.75	0.33	0.00	0.00	0.00	0.18
08. Using smartphone outside	0.32	0.00	**0.45**	0.00	0.00	0.00	0.00	0.00	0.00
09. Wearing a red plaid shirt	**0.58**	0.44	0.25	0.00	0.12	0.00	0.00	0.00	0.00
10. Having a meeting in China	**1.00**	0.25	0.45	0.00	0.57	0.00	0.40	0.25	0.32
Average Score	**0.61**	**0.44**	**0.40**	**0.29**	**0.25**	**0.19**	**0.12**	**0.06**	**0.13**

answering. In: CVPR (2018)

2. Dang-Nguyen, D.T., Piras, L., Riegler, M., Boato, G., Zhou, L., Gurrin, C.: Overview of imagecleflifelog 2017: Lifelog retrieval and summarization. In: CLEF (Working Notes) (2017)
3. Dang-Nguyen, D.T., Piras, L., Riegler, M., Tran, M.T., Zhou, L., Lux, M., Le, T.K., Ninh, V.T., Gurrin, C.: Overview of ImageCLEFlifelog 2019: Solve my life puzzle and Lifelog Moment Retrieval. In: CLEF2019 Working Notes. CEUR Workshop Proceedings, CEUR-WS.org <http://ceur-ws.org>, Lugano, Switzerland (September 09-12 2019)
4. Dang-Nguyen, D.T., Piras, L., Riegler, M., Zhou, L., Lux, M., Gurrin, C.: Overview of imagecleflifelog 2018: daily living understanding and lifelog moment retrieval. In: CLEF2018 Working Notes (CEUR Workshop Proceedings). CEUR-WS. org¡ http://ceur-ws. org¿, Avignon, France (2018)
5. Dao, M., Kasem, A., Nazmudeen, M.S.H.: Leveraging content and context in understanding activities of daily living. In: Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum, Avignon, France, September 10-14, 2018. (2018)
6. Dao, M.S., Nguyen, D., Tien, D., Riegler, M., Gurrin, C.: Smart lifelogging: recognizing human activities using phasor (2017)
7. Dimiccoli, M., Cartas, A., Radeva, P.: Activity recognition from visual lifelogs: State of the art and future challenges. In: Multimodal Behavior Analysis in the Wild, pp. 121–134. Elsevier (2019)
8. Gurrin, C., Joho, H., Hopfgartner, F., Zhou, L., Albatal, R.: Overview of ntcir-12 lifelog task. In: Kando, N., Kishida, K., Kato, M.P., Yamamoto, S. (eds.) Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies. pp. 354–360 (2016)
9. Gurrin, C., Joho, H., Hopfgartner, F., Zhou, L., Gupta, R., Albatal, R., Nguyen, D., Tien, D.: Overview of ntcir-13 lifelog-2 task. NTCIR (2017)

10. Harvey, M., Langheinrich, M., Ward, G.: Remembering through lifelogging. Pervasive Mob. Comput. **27**(C), 14–26 (Apr 2016). https://doi.org/10.1016/j.pmcj.2015.12.002, http://dx.doi.org/10.1016/j.pmcj.2015.12.002
11. Ionescu, B., Müller, H., Péteri, R., Cid, Y.D., Liauchuk, V., Kovalev, V., Klimuk, D., Tarasau, A., Abacha, A.B., Hasan, S.A., Datla, V., Liu, J., Demner-Fushman, D., Dang-Nguyen, D.T., Piras, L., Riegler, M., Tran, M.T., Lux, M., Gurrin, C., Pelka, O., Friedrich, C.M., de Herrera, A.G.S., Garcia, N., Kavallieratou, E., del Blanco, C.R., Rodríguez, C.C., Vasillopoulos, N., Karampidis, K., Chamberlain, J., Clark, A., Campello, A.: ImageCLEF 2019: Multimedia retrieval in medicine, lifelogging, security and nature. In: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the 10th International Conference of the CLEF Association (CLEF 2019), LNCS Lecture Notes in Computer Science, Springer, Lugano, Switzerland (September 9-12 2019)
12. Patterson, G., Xu, C., Su, H., Hays, J.: The sun attribute database: Beyond categories for deeper scene understanding. Int. J. Comput. Vision **108**(1-2), 59–81 (May 2014). https://doi.org/10.1007/s11263-013-0695-z, http://dx.doi.org/10.1007/s11263-013-0695-z
13. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2017)

Algorithm 1 Create a set of time-ordered-associative (TOU) images

Input: $Q = \{q[k]\}$, $D = \{d[l]\}$, θ, α (#days), β (#clusters)
Output: $P_{TOU}[i][j]$

$P_{TOU}[i][j] \Leftarrow \emptyset$
for i=1..α **do**
  for j=1..β **do**
    Filter D so that only images taken on day i within cluster j is enable
    Establish $P = \{p(iD, iQ)[m]\}$ so that $\forall m < n : time(d[p[m].iD]) < time(d[p[n].iD]) \wedge similarity(d[p[m].iD], q[p[m].iQ]) > \theta$ **and** $\forall m \neq n : p(iD, iQ)[m] \neq p(iD, iQ)[n]$
    repeat
      $\forall n \leq \|P\|$
      if (p[n].iQ == p[n+1].iQ) **then**
        if $(similarity(d[p[n].iD], q[p[n].iQ]) > similarity(d[p[n+1].iD], q[p[n].iQ]))$ **then**
          Delete $p[n+1]$ **else** Delete $p[n]$
        end if
        Rearrange the index of P
      end if
    until cannot delete any item of p
    $P_{TOU}[i][j] \Leftarrow P$ (P must satisfy $\forall n : p[n].iQ \neq p[n+1].iQ$)
  end for
end for
return $P_{TOU}[i][j]$

Algorithm 2 Create a set of unique-time-ordered-associative (UTOU) images

Input: $Q = \{q[k]\}$, $D = \{d[l]\}$, $P_{TOU}[i][j]$, α (#days), β (#clusters)
Output: $P_{UTOU}[i][j][z]$

1: $slidingWindow = \|Q\|$
2: **for** $i = 1..\alpha$ **do**
3: $z = 1$, $N = \|P_{TOU}[i][j]\|$
4: **for** $j = 1..\beta$ **do**
5: **for** $n = 1..(N - slidingWindow)$ **do**
6: $substr \Leftarrow P_{TOU}[i][j][m]_{m=n..(n+slidingWindow)}$
7: **if** $(\forall m \neq n : substr[m].iQ \neq substr[n].iQ$ **and** $\|substr\| == slidingWindow)$ **then**
8: $P_{UTOU}[i][j][z] \Leftarrow substr$, $z \Leftarrow z + 1$
9: **end if**
10: **end for**
11: **end for**
12: **end for**
13: **return** $P_{UTOU}[i][j][z]$

Algorithm 3 Find the BEST set of UTOU images

Input: $Q = \{q[k]\}$, $D = \{d[l]\}$, $P_{UTOU}[i][j][z]$, α (#days), β (#clusters)
Output: P_{BEST}

1: **for** $j = 1..\beta$ **do**
2: $CoverSet \Leftarrow \emptyset$
3: **for** $i = 1..\alpha$ **do**
4: $CoverSet[i] \Leftarrow CoverSet \cup P_{STOAU}[i][j]$
5: **end for**
6: Find the most common pattern *pat* from all patterns contained in $CoverSet$
7: Delete all subsets of $CoverSet$ that do not match *pat*
8: For each remained subset of $CoverSet$, calculate the average similarity with Q
9: Find the subset $CoverSet_{largest}$ that has the largest average similarity with Q
10: $P_{BESTCLUSTER}[j] \Leftarrow CoverSet_{largest}$
11: **end for**
12: Find the subset P_{BEST} of $P_{BESTCLUSTER}[j]$ that has the largest average similarity with Q
13: **return** P_{BEST}

Algorithm 4 A Interactive Watershed-based Lifelog Moment Retrieval

Input: QT, $BoW_{Concepts}$, $\{I_i\}_{i=1..N}$
Output: $LMRT$

{OFFLINE}
$\{BoW_{I_i}\}_{i=1..N} \Leftarrow \emptyset$
$\{(Vc_i, Va_i)\}_{i=1..N} \Leftarrow \emptyset$
$\forall i \in [1..N], BoW_{I_i} \Leftarrow I2BoW(I_i)$
$\forall i \in [1..N], (Vc_i, Va_i) \Leftarrow featureExtractor(I_i)$
$\{C_m\} \Leftarrow cluster(\{I_i\}_{i=1..N})$ using $\{(Vc_i, Va_i)\}$
{ONLINE}
$BoW_{QT}^{Noun} \Leftarrow Pos\_tag(QT).Noun$
$BoW_{QT}^{Aug} \Leftarrow GoogleSearchAPI(BoW_{QT}^{Noun})$
$BoW_{QT}^{Noun} \Leftarrow Filter(BoW_{QT}^{Aug} \cap BoW_{Concepts})$
$BoW_{QT} \Leftarrow Interactive(BoW_{QT}^{Noun}, Pos\_tag(QT))$
$\left\{Seed_j^1\right\} \Leftarrow Interactive(Query(BoW_{QT}, \{BoW_{I_i}\}_{i=1..N}))$
$LMRT^1 \Leftarrow \left\{C_k | \forall j \in \left\|\left\{Seed_j^1\right\}\right\|, Seed_j^1 \in \{C_k\}\right\}$
$\{C_l^{rem}\} \Leftarrow \{C_m\} - LMRT^1$
$\left\{Seed_j^2\right\} \Leftarrow Query(\left\{Seed_j^1\right\}, \{C_l^{rem}\})$
$LMRT^2 \Leftarrow \left\{C_l | \forall j \in \left\|\left\{Seed_j^2\right\}\right\|, Seed_j^2 \in \{C_l^{rem}\}\right\}$
$LMRT \Leftarrow LMRT^1 \cup LMRT^2$
$LMRT \Leftarrow Intearactive(LMRT)$
return $LMRT$

Algorithm 5 cluster

Input: $I = \{I_i\}_{i=1..N}, F = \{(Vc_i, Va_i)\}_{i=1..N}, \theta_a, \theta_b$
Output: $\{C_k\}$

$k \longleftarrow 0$
$C = \{C_k\} \longleftarrow \emptyset$
$I^{temp} \Leftarrow SORT_{bytime}(I, F)$
repeat
  $v_a \longleftarrow I^{temp}.F.Va[0]$
  $v_c \longleftarrow I^{temp}.F.Vc[0]$
  $C[0] \Leftarrow \cup I^{temp}.I[0]$
  $I^{temp} \Leftarrow I^{temp} - I^{temp}[0]$
  for i=1..$\left\|I^{temp}\right\|$ **do**
    if $(similarity(v_a, I^{temp}.F.Va[i]) > \theta_a$ **and** $similarity(v_c, I^{temp}.F.Vc[i]) > \theta_c$ **then**
      $C[k] \Leftarrow \cup I^{temp}.I[i]$
      $I^{temp} \Leftarrow I^{temp} - I^{temp}[i]$
    else
      $k \longleftarrow k + 1$
      Break
    end if
  end for
until $\left\|I^{temp}\right\| == 0$
return C