An interactive atomic-cluster watershed-based system for lifelog moment retrieval

Van-Luon Tran[1], Trong-Dat Phan[1], Anh-Vu Mai-Nguyen[1], Anh-Khoa Vo[1], Minh-Son Dao*[2], and Koji Zettsu[2]

[1] University of Science, VNU-HCMC, Vietnam
{1612362,1512102,1612904,1512262}@student.hcmus.edu.vn
[2] National Institute of Information and Communications Technology, Japan
{dao,zettsu}@nict.go.jp

Abstract. In this paper, we introduce a new interactive atomic-cluster watershed-based system for lifelog moment retrieval. We investigate three essential components that can help improve accuracy and support both amateur and professional users to enhance their querying based on different content and context hypothesis. These components are (1) the atomic cluster function that clusters dataset to a set of time-consecutive images that shares the same content and context constraints, (2) the text-to-sample image generation that helps to overcome the gap between textual queries of users and visual-based feature vectors database, and (3) The interactive interface that assists users to imagine what they want to look for better. The system is customized to meet the challenge of lifelog moment retrieval of imageCLEFlifelog2020. The evaluation and comparison of our method to others confirm the stability of our method when people want to retrieve a large number of results within 100 top results.

1 Introduction

Finding a moment in our past with a few hints or cues is the activity we probably carry on almost every day. Except for extraordinary people who have a fantastic memory that can recall every moment in their lives within a split-second, ordinary people need more time to narrow down their searching scope from a very abstract level to detail. The same situation happens when people want to find their historical moment from their lifelog data. That leads to the fact that if people can have an interactive system that can help them turn their queries from an amateur sketch to an artist's paint, they will retrieve their moment faster and more precisely [1], [2]

Besides, turning a few keywords and less semantic contents of users' text queries to somethings that can be understood by the search engine is another

* corresponding author

challenge [3]. There is still a big gap between the natural language spoken by users and machine language designed for search engines [4] that can prevent the improvement of accuracy. Feature selection is another factor that can assist in bridging this gap [5], [6], and in support of the well-organized dataset.

Based on the discussion mentioned above, we design an interactive atomic-cluster watershed-based system for lifelog moment retrieval. This system is customized to meet the lifelog moment retrieval (LMRT) challenge of imageCLEFlifelog2020 [7], a lab task of imageCLEF2020 [8].

Our system's main contributions are:

1. We introduce an atomic cluster, a set of time-consecutive images that shares the same content and context constraints.
2. We build the text-to-sample image generation to overcome the gap between textual queries of users and visual-based feature vectors database.
3. We create an interactive interface to help users imagine what they want to look for better.

We organize this paper as follows: Section 2 describes our method in details, Section 3 discusses the challenge and evaluates our results, and Section 4 concludes our paper and sketches our plan in the future.

2 Methodology

The principal idea of our method bases on the following observations:

1. daily activities of people can be divided into sequential atomic moments whose content has a consensus of both content and context. In other words, lifelog data recorded during a day can be divided into sequential atomic clusters whose content reflects a unique semantic meaning with a consensus of spatiotemporal dimension. These atomic clusters cannot be divided into smaller clusters. Hence, if we could find one image that matches the query (i.e., seeds), we can count in the atomic cluster the image belongs and the neighbors of the atomic cluster (i.e., watershed).
2. people can decide which data do not satisfy their queries. In other words, people can remove irrelevant data and modify their queries to get more relevant data. Hence, if we can provide people a friendly interface for interactive querying, people can improve the qualification of the querying system.

We call the system built upon these observations is an interactive atomic-cluster watershed system. The system has four vital components (1) atomic-cluster clustering (*Cluster* function), (2) text-to-sample image generation (*Attention* function), (3) querying by text-to-sample images (*Query* function), (4) interaction (*Interactive* function), and (5) querying by user's images (*Query* function). Algorithm 1 and Figure 1 describe and illustrate how the system works, respectively.

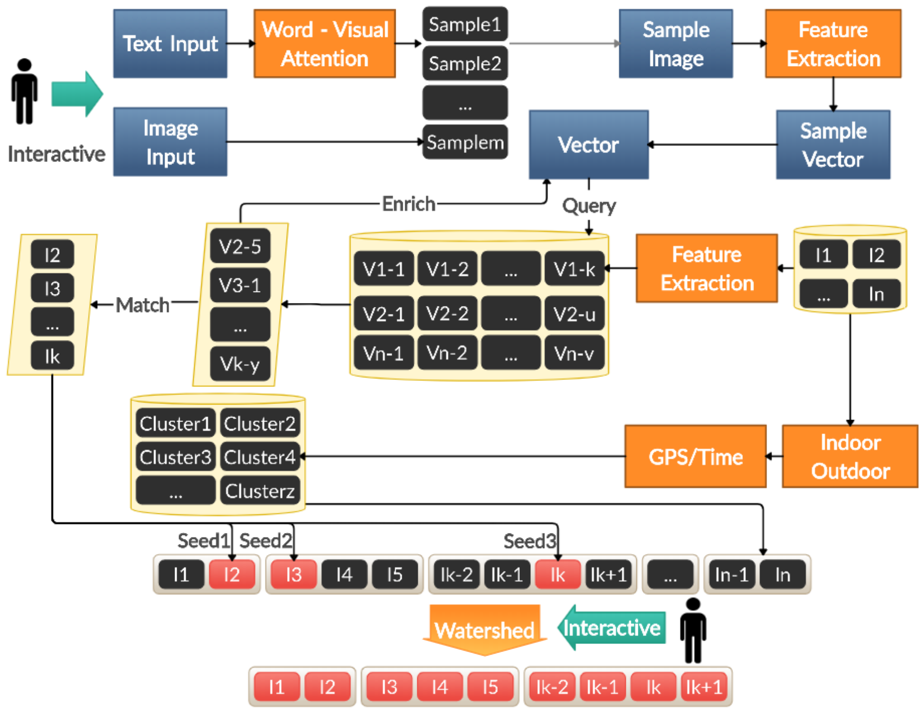

Fig. 1: An Interactive Multimodal Lifelog Retrieval System

2.1 Notations and definitions

There are several notations and feature vectors which are used for our system. Thus, we define them below:

- Let Q denote the query sentence.
- Let $I = \{I_i\}_{i=1..N}$ denote the set of given images (e.g., lifelog).
- Let $C = \{C_k\}$ denote a set of atomic clusters.
- Let S denote the set of samples for each query, and S^i denote the status of set S at the time i.
- Let $BoV = \{V_{i-k}\}_{i=1..N}^{k=1..m}$ denote the set of feature vectors of objects extracted from I, where V_{i-k} denotes the feature vector of the k^{th} object of I_i and BoV_i denotes the set of all object vectors of I_i.
- Let BoV_{DB} denote the database stores all object vectors of all images in I.
- Let $Seed$ and $LMRT$ denote a set of seeds and lifelog moments, respectively.
- Let denote $\boldsymbol{V}_{o_i}$ is the 1024-D vector representation of the i^{th} object region in the photos.
- Let denote $\boldsymbol{p}_i$ is output vector of the i^{th} image.
- Let denote $\boldsymbol{V}_{w_i}$ is word embedding vector of the i^{th} word.

Algorithm 1 Query-to-Sample Attention-based Search Engine

```
Input: $Q$, $\{I_i\}_{i=1..N}$
Output: $LMRT$
    {ONLINE}
 1: $Sample \Leftarrow \emptyset$
 2: if $type(Q) ==$ "$text$" then
 3:     $Sample \cup Word - VisualAttention(Q, I_{external})$
 4: end if
 5: if $type(Q) ==$ "$image$" then
 6:     $Sample \cup Q$
 7: end if
 8: $V_{sample} \Leftarrow FE(Sample)$
    {OFFLINE}
 9: $\{C_m\} \Leftarrow Cluster(\{I_i\}_{i=1..N})$
10: $BoV \Leftarrow \emptyset$
11: $\forall i \in [1..N], BoV_{I_i} \Leftarrow FE(I_i)$
12: $BoV_{DB} \Leftarrow FAISS(BoV)$
    {ONLINE}
13: $S \Leftarrow \emptyset$
14: $S^0 = S \Leftarrow S \cup \{V_{sample}\}$
15: while $S^i \neq S^{i+1}$ do
16:     $S \Leftarrow S \cup Query(S, BoV_{DB})$
17:     $i \Leftarrow i + 1$
18: end while
19: $seed \Leftarrow \{I_i | \forall V_{ik} \in S\}$
20: $LMRT \Leftarrow \{C_k | \forall j \in \|Seed\|, Seed_j \in \{C_k\}\}$
21: $LMRT \Leftarrow Interactive(LMRT)$
22: return $LMRT$
```

2.2 System Workflow

In this subsection, we give a detailed explanation about our Interactive Multimodal Lifelog Retrieval System depicted in figure 1 and the Algorithm 1.

There are two stages in our system's workflow: (1) offline stage, and (2) online stage.

The former stage is for data preprocessing. Firstly, we divide lifelog images into atomic clusters by utilizing *Clustering function*, described in 2.4. Then, all lifelog images are converted to V_{sample} by using *Feature Extraction* (FE) function, described in 2.3. In other words, V_{sample} contains feature vectors extracted from images. To make use full of FAISS [9], we embed these V_{sample} into a unified database by applying FAISS's function.

The latter stage is for textual and visual querying. For textual querying, our system activates *Attention function*, described in 2.5, to generate sample images from texts. Then, sample images (and input images if users carry on visual querying) are fed into the FE function to create related V_{sample}. The V_{sample} is used to find the most similar feature vectors from the FAISS-based database with the predefined similarity threshold. Next, we enrich V_{sample} by adding these found feature vectors and re-querying upon FAISS-based databased until no new feature vectors found. All images that have their features vectors appear in this set are considered as the queried results and set as seeds. The final results are all atomic clusters contained these seeds. Then, users use *Interactive* tools described in 2.6 to polish the output, so they receive wanted results.

2.3 Feature Extraction

- $\boldsymbol{V}_{o_i}$ is extracted by using object detection model (Faster-RCNN backbone Resnet) in scaled Visual Genome dataset [10] (removing semantic overlapping classes)
- $\boldsymbol{p}_i$ is extracted by utilizing place detection model described in 2.4
- $\boldsymbol{V}_{w_i}$ is built as follows: Hidden state 768-D vectors extracted from BERT [11] are combined with one linear Conditional Random Field layer to construct seq2seq model [12] and output keywords (from a long input query sentence) with their representation vectors.

2.4 Atomic-Cluster Clustering

As mentioned in previous sections, an atomic cluster contains a set of consecutive lifelog images (and related metadata) whose content reflects a particular activity constrained by location, time, and semantic meaning. We have the whole dataset clustered into atomic clusters by two steps (1) enhance the quality of metadata and (2) cluster multimodal data. The former applies a self-supervised learning method to regenerate metadata. By utilizing SimCLR method [13], we manually label place names for about 20k images and then train a new model to label the remaining images in a dataset automatically. Finally, we strengthen metadata's location constraints by having more precise place names than the original metadata. The latter utilizes the updated metadata and feature vectors extracted from images as the input of the clustering method proposed in [14] to form atomic clusters.

2.5 Text-to-sample Image Generation

The essential idea of this function is to replace a textual query with a set of visual queries. First, we create a dataset of objects using open image datasets (e.g., COCO, image365). It means that we have a set of object names, and each object name links to a set of images that contains the object. Then, we parse a textual query to extract the object's names replaced by linked images. Notably, we utilized the attention mechanism [15] to build our function, as described in Algorithm 2. We firstly utilize Top-Down Attention LSTM in a two-layer LSTM model for captioning images from feature vectors of regions detected by the object detection model [16]. We then determine a useful feature transformation from word vector space to visual space using a well-trained Bottom-up Attention model.

Algorithm 2 Text-to-sample Image Generation

```
Input: Word set ${Word_i}_{i=1..M}$, Object set ${Obj_j}_{j=1..N}$
Output: ${Word_i : Obj_j}$ map from word to relevant object.
 1: ${V_{w_i}}_{i=1..M} \Leftarrow WordEmb(Word)$
 2: $\left\{V_{o_j}\right\}_{j=1..N} \Leftarrow FE(Obj)$
 3: Training bottom-up attention model as in [15].
 4: for all $k \leq \|V_w\|$ do
 5:     $\hat{v}_k \Leftarrow \sum_{j=1}^{N} \alpha_{k,j} v_j$
 6:     $j' \Leftarrow \arg\max_j \alpha_{kj}$
 7:     $\hat{v}_k \Leftarrow v_{j'}$
 8:     $\hat{v}_k$ is the optimized presentation for $Word_k$ in visual space
 9: end for
10: return ${Word_i : Obj_j}$ where $i = 1..M, j = 1..N$
```

2.6 Interaction

After having the first results generated by the query by sample function, users can filter the results using other metadata such as visible objects, places, and time. These metadata are saved as text files using PostgreSQL and stored in Logic Server, as described in 2.7. Besides, users can re-query by manually selecting samples from results visualized on the system's interface or add more query categories by texts. Moreover, users can delete inappropriate images as they think. These images are taken into account by the system to mark as outliers or unnecessary items for the next query. Algorithm 3 explained how the interaction works.

Algorithm 3 Interactive Algorithm

```
Input: PostgreSQL for metadata $P$,
    $I \Leftarrow LMRT$
Output: $I$
 1: while Interactive do
 2:     $I \Leftarrow Remove(I)$ {Continue if there is no removed image}
 3:     $F \Leftarrow$ Input filters
 4:     $I \Leftarrow P.select(I, F)$ {Continue if $F$ is none}
 5:     $I \Leftarrow I \cup$ Re-query(input images or text)
 6: end while
 7: return $I$
```

2.7 Interactive System Architecture

To build a flexible system, we design our system following three-tier and three-layer architecture, depicted in figure 2. The first layer is the presentation layer

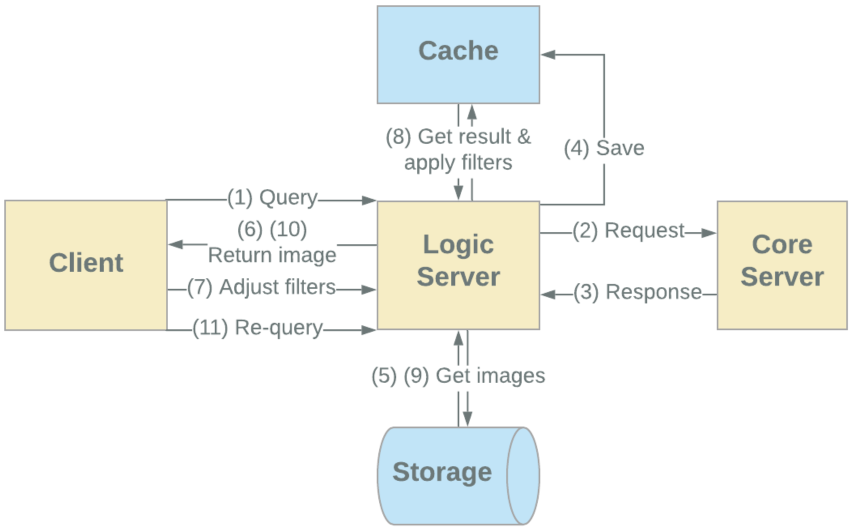

Fig. 2: The System Architecture

on a User Client, the second is the logic layer on Logic Server, and the last one is the core layer on the Core Server.

At the first layer, it is a convenient web-based interface where users can interact with our system. This interface can easily be installed in a wide range of operating systems. It firstly allows users to type text queries, select filters, and input sample images, which is a powerful tool for users to describe which images they would like to retrieve. Then, these data, along with IDs of removed images (in the case users delete the queried results of the previous interaction), will be pushed to Logic Server. Next, the interface has responsibility for presenting images sent from Logic Server. Before users re-query, they can modify their text query, adjust filters, choose images from other sources, and remove unwanted images. They can re-query until presented images satisfy user's demand. Finally, users use the export function to download images or image IDs.

At the second layer, Logic Server has responsibility for processing requests from User Client. Firstly, this server converts query to a suitable form and send it to Core Server. Then, Logic Server receives outputted results with IDs of images and IDs of related atomic clusters. The result will be saved directly to Cache, a temporary memory on Logic Server. At the following steps, depending on the type of filters, this server will apply the filters on the whole dataset or only the results stored in Cache. There are two types of filter (1) Extend Filter and (2) Narrow Filter. With the former, Logic Server will find all images whose metadata are matched to this filter before adding these image's IDs to Cache. With the latter, from IDs in Cache, Logic Server will select images whose metadata is fitted to the filter. Finally, the server returns filtered images and ranked clusters to User Client.

In terms of Core Server, it receives input from Logic Server and sends result reversely after completely processing. Core Server is an always-on server where AI components are deployed.

3 Experiments

In this section, we present our system's experiment results when applying to the lifelog dataset CLEF2020.

3.1 Dataset and Evaluation Metrics

The CLEF2020 dataset has been captured by one active lifelogger for 114 days between 2015 and 2018. It contains not only over 191000 lifelog images also metadata, including visual concepts, attributes, semantic content, to name a few. The training set has ten topics, and each topic is described by title and description. These titles are: (1) Having beers in a bar, (2) Building Personal Computer, (3) In A Toy Shop, (4) Television Recording, (5) Public Transport In Home Country, (6) Seaside Moments, (7) Grocery Stores, (8) Photograph of The Bridge, (9) Car Repair, (10) Monsters. The topic descriptions are used to explain in detail about the content and context of each query. Similar to the training set, the testing set has ten topics which are: (1) Praying Rite, (2) Recall, (3) Bus to work - Bus to home, (4) Bus at the Airport, (5) Medicine cabinet, (6) Order Food in the Airport, (7) Seafood at Restaurant, (8) Meeting with people, (9) Eating Pizza, (10) Socialising.

The evaluation metrics are defined by ImageCLEFliflog 2020 as follow:

– Cluster Recall at X (CR@X) - a metric that assesses how many difference 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.

3.2 Evaluation and Comparison

The ImageCLEFlifelog challenge has five participant teams including: (1) RRibeiro, (2) FatmaBA_RegimLab, (3) DCU_Team, (4) BIDAL_HCMUS (ourselves), (5) HCMUS. We are ranked in the second position. Table 1 and 2 shows our results running on the training and testing set while table 3 and 4 denote the comparison to the other teams.Figures 3-12 illustrate our results of the testing stage.

Table 1: Results running on CLEF2020 Training Set

Query	Run 1			Run 2		
	P@10	CR@10	F1@10	P@10	CR@10	F1@10
1	0.7	0.50	0.58	1.00	0.75	0.86
2	0.60	0.50	0.55	1.00	0.50	0.67
3	0.30	0.50	0.38	0.70	1.00	0.82
4	0.70	0.67	0.68	1.00	0.67	0.80
5	0.20	0.11	0.14	0.90	0.44	0.60
6	0.60	0.50	0.55	0.80	0.50	0.62
7	0.50	0.44	0.47	0.80	0.78	0.79
8	0.30	0.50	0.38	0.40	0.50	0.44
9	0.60	1.00	0.75	0.80	1.00	0.89
10	0.30	1.00	0.46	0.30	1.00	0.46

Table 2: Results running on CLEF2019 Testing Set

Query	Run 9			Run 10		
	P@10	CR@10	F1@10	P@10	CR@10	F1@10
1	1	1	1	1	1	1
2	0.60	0.50	0.55	0.60	0.50	0.55
3	0.50	1	0.67	0.50	1	0.67
4	0.70	0.50	0.58	0.80	0.50	0.62
5	0.70	0.78	0.74	0.70	0.78	0.74
6	1	0.33	0.50	1	0.50	0.67
7	1	0.20	0.33	0.80	0.60	0.69
8	0.60	1	0.75	0.60	1	0.75
9	0.90	0.67	0.77	0.90	0.67	0.77
10	0.50	0.50	0.50	0.50	0.50	0.50

Table 3: Comparison to the other teams (F1@10 metric)

Query	F1@10					
	Baseline	FatmaBA_RegimLab	RRibeiro	DCU_Team	BIDAL_HCMUS	HCMUS
1	**1**	0.58	0.95	0.50	**1**	**1**
2	0.4	0	0	0.22	0.55	**0.86**
3	0.17	0.29	0.67	0.95	0.67	**1**
4	0	0.14	0	0.27	**0.62**	0.55
5	0.21	0	0.77	0.68	0.74	**0.83**
6	0.13	0.13	0.50	0.25	0.67	**0.68**
7	0.24	0	0.67	0.44	**0.69**	0.57
8	0	0	**0.82**	0.33	0.75	0.67
9	0.57	0.75	0.80	0.68	0.77	**0.95**
10	0.50	0	0	0.50	0.50	**1**
Avg	0.32	0.19	0.52	0.48	0.69	**0.81**

Table 4: Comparison to the other teams (F1@50 metric)

Query	F1@50					
	Baseline	FatmaBA_RegimLab	RRibeiro	DCU_Team	BIDAL_HCMUS	HCMUS
1	0.63	0.46	**0.99**	0.17	0.33	0.33
2	0.49	0.03	0	**0.77**	0.19	0.32
3	0.04	0.34	**0.95**	0.31	0.65	0.33
4	0	0.07	0	0.16	**0.39**	0.32
5	0.17	0	0.65	0.62	0.61	**0.70**
6	0.04	0.04	0.17	0.29	**0.38**	0.23
7	0.09	0	**0.85**	0.50	0.67	0.27
8	0	0.04	**0.97**	0.08	0.80	0.18
9	0.18	0.49	**0.76**	0.28	0.60	0.37
10	0.50	0.53	0	0.50	0.50	**1**
Avg	0.21	0.15	**0.53**	0.37	0.51	0.41

When comparing the results evaluated by F1@10 and F1@50 metrics, we found that our scores are less fluctuation than the others (some other teams have a massive reduction in their scores), as described in table 3 and 4. That probably could lead to the conclusion that our proposed method is stable, especially if the user wants to retrieve a large of images.

In some queries, we have worse scores because of misunderstanding the content and context of queries. For instance, query 5 has the title: 'Medicine cabinet' and description: 'Find the moment when u1 was looking inside the medicine cabinet in the bathroom at home', we very confused when trying to confirm whether the lifelogger really looks inside the medicine cabinet or appear nearby (i.e., the medicine cabinet is captured by lifelog camera, but the u1 does not look at it). The result of query 5 is shown in figure 7.

Furthermore, we found that the ground truth could have some incorrect points. We have verified with the organizers that the ground-truth might not be precise. For example, the image ID b00000986_21i6bq_20150225_161718e (in query 9) and the image ID 20160904_120624_000 (in query 5) should have been in the ground-truth. Figures 7 and 11 illustrate the results of queries 5 and 9, where the red rectangle denotes mentioned images. That probably makes our results not precise enough as we expected.

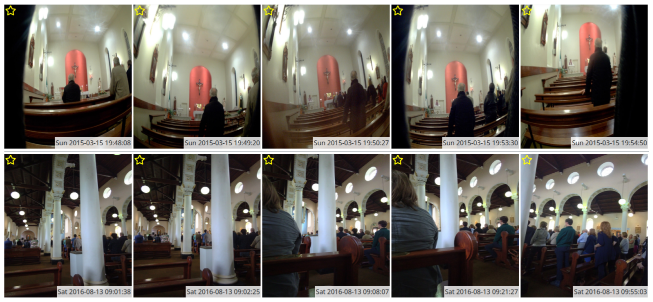
Fig. 3: The top ten results of query 1 ”Praying Rite” (F1@10 = 1)

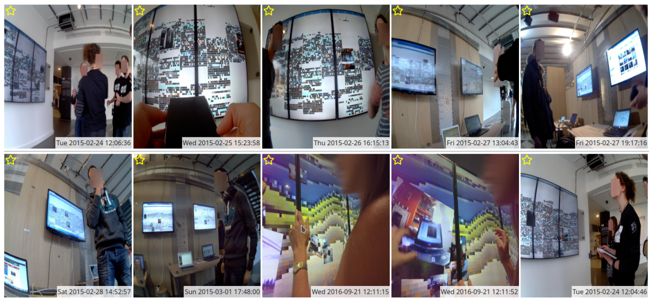
Fig. 4: The top ten results of query 2 ”Recall” (F1@10 = 0.55)

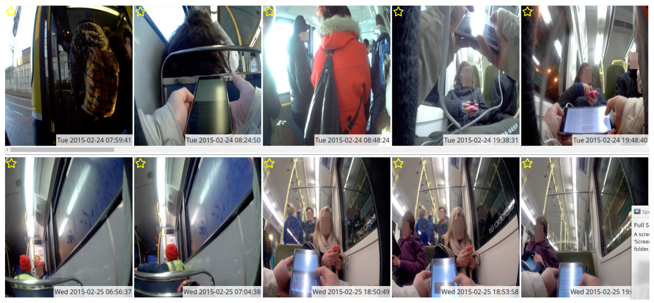
Fig. 5: The top ten results of query 3 ”Bus to work - Bus to home” (F1@10 = 0.67)

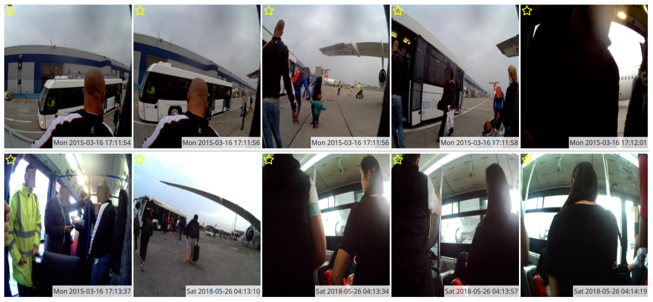
Fig. 6: The top ten results of query 4 ”Bus at the airport” (F1@10 = 0.62)

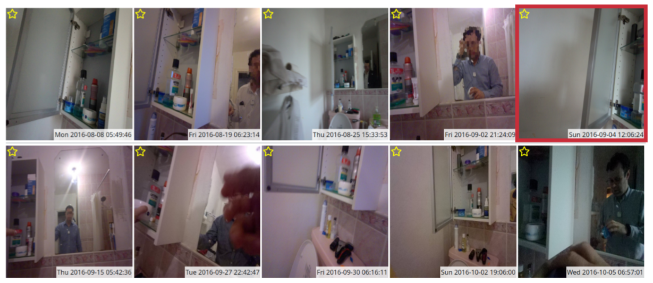
Fig. 7: The top ten results of query 5 ”Medicine cabinet” (F1@10 = 0.74)

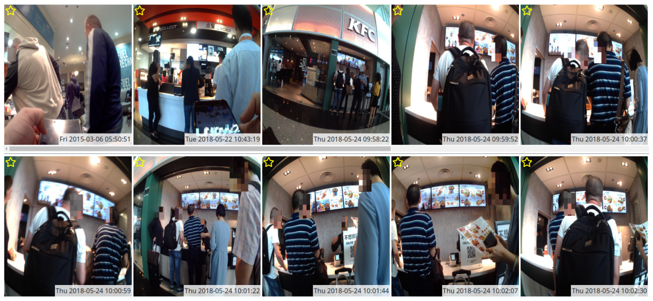
Fig. 8: The top ten results of query 6 ”Order Food in the Airport” (F1@10 = 0.67)

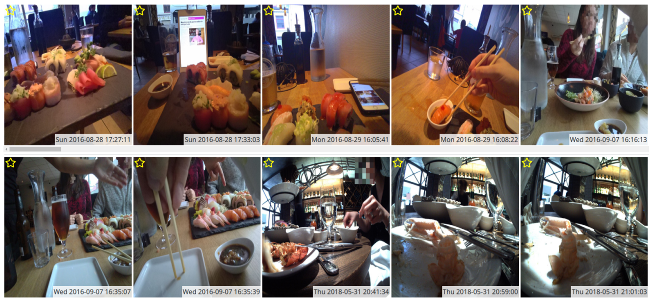

Fig. 9: The top ten results of query 7 ”Seafood at Restaurant” (F1@10 = 0.69)

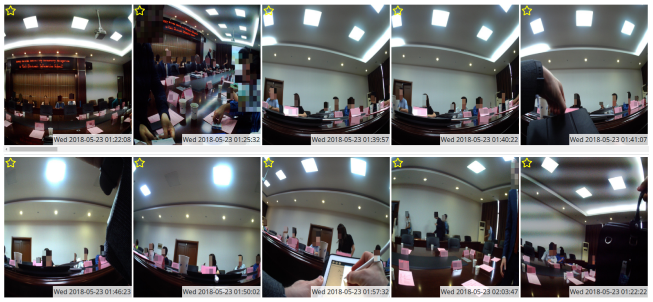

Fig. 10: The top ten results of query 8 ”Meeting with people” (F1@10 = 0.75)

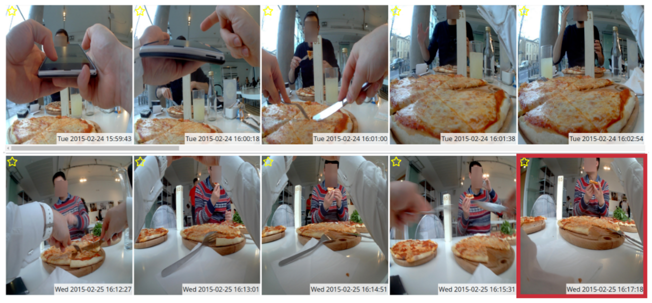

Fig. 11: The top ten results of query 9 ”Eating Pizza” (F1@10 = 0.77)

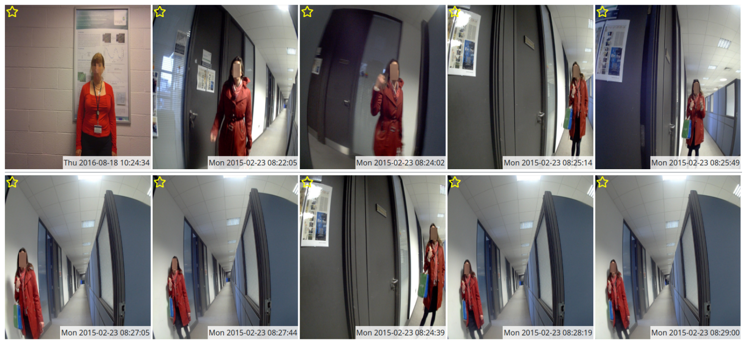

Fig. 12: The top ten results of query 10 "Socialising" (F1@10 = 0.50)

4 Conclusions and Future Works

We introduced a new interactive atomic-cluster watershed-based system for lifelog moment retrieval. The system is specially customized to meet the requirement of the imageCLEFlifelog2020 challenges. The system first indexes the database based on atomic clusters that contain similar data based on our similarity measure. The reason behind the atomic clusters is that whenever one image is found, its atomic cluster counts in. We store feature vectors extracted from data in FAISS database for further querying. We convert all textual queries into visual queries by using the attention mechanism approach. The system provides a friendly interactive interface that allows users to select precise results and re-query with modification. Our results are evaluated and compared to other participants with positive accuracy. We will investigate the atomic clustering function to improve the consensus and compact of atomic clusters in the future. Moreover, we will consider wrapping spatiotemporal information to the querying engine by strengthening semantic constraints. Last but not least, we will focus on feature engineering and similarity measures to have a higher accuracy of querying.

Acknowledgement

This research is conducted under the Collaborative Research Agreement between National Institute of Information and Communications Technology and University of Science, Vietnam National University at Ho-Chi-Minh City.

References

1. S. Park, J. Song, M. Park, and Y. M. Ro, "Ivist: Interactive video search tool in vbs 2020," in *International Conference on Multimedia Modeling*. Springer, 2020, pp. 809–814.

2. B. Jónsson, O. S. Khan, D. C. Koelma, S. Rudinac, M. Worring, and J. Zahálka, "Exquisitor at the video browser showdown 2020," in *MultiMedia Modeling*, Y. M. Ro, W.-H. Cheng, J. Kim, W.-T. Chu, P. Cui, J.-W. Choi, M.-C. Hu, and W. De Neve, Eds. Cham: Springer International Publishing, 2020, pp. 796–802.
3. S. Andreadis, A. Moumtzidou, K. Apostolidis, K. Gkountakos, D. Galanopoulos, E. Michail, I. Gialampoukidis, S. Vrochidis, V. Mezaris, and I. Kompatsiaris, "Verge in vbs 2020," in *International Conference on Multimedia Modeling*. Springer, 2020, pp. 778–783.
4. L. Chen, S. Shang, C. Yang, and J. Li, "Spatial keyword search: a survey," *Geoinformatica*, vol. 24, p. 85–106, 2020.
5. R. Zhang, F. Nie, X. Li, and X. Wei, "Feature selection with multi-view data: A survey," *Information Fusion*, vol. 50, pp. 158 – 167, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1566253518303841
6. R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki, "A survey on semi-supervised feature selection methods," *Pattern Recognition*, vol. 64, pp. 141 – 158, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0031320316303545
7. V.-T. Ninh, T.-K. Le, L. Zhou, L. Piras, M. Riegler, P. l Halvorsen, M.-T. Tran, M. Lux, C. Gurrin, and D.-T. Dang-Nguyen, "Overview of ImageCLEF Lifelog 2020:Lifelog Moment Retrieval and Sport Performance Lifelog," in *CLEF2020 Working Notes*, ser. CEUR Workshop Proceedings. Thessaloniki, Greece: CEUR-WS.org <http://ceur-ws.org>, September 22-25 2020.
8. B. Ionescu, H. Müller, R. Péteri, A. B. Abacha, V. Datla, S. A. Hasan, D. Demner-Fushman, S. Kozlovski, V. Liauchuk, Y. D. Cid, V. Kovalev, O. Pelka, C. M. Friedrich, A. G. S. de Herrera, V.-T. Ninh, T.-K. Le, L. Zhou, L. Piras, M. Riegler, P. l Halvorsen, M.-T. Tran, M. Lux, C. Gurrin, D.-T. Dang-Nguyen, J. Chamberlain, A. Clark, A. Campello, D. Fichou, R. Berari, P. Brie, M. Dogariu, L. D. Ştefan, and M. G. Constantin, "Overview of the ImageCLEF 2020: Multimedia retrieval in lifelogging, medical, nature, and internet applications," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, ser. Proceedings of the 11th International Conference of the CLEF Association (CLEF 2020), vol. 12260. Thessaloniki, Greece: LNCS Lecture Notes in Computer Science, Springer, September 22-25 2020.
9. J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with gpus," *arXiv preprint arXiv:1702.08734*, 2017.
10. R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," 2016. [Online]. Available: https://arxiv.org/abs/1602.07332
11. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
12. I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.
13. T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *arXiv preprint arXiv:2002.05709*, 2020.
14. T. Phan, M. Dao, and K. Zettsu, "An interactive watershed-based approach for lifelog moment retrieval," in *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*, Sep. 2019, pp. 282–286.

15. P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.
16. S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.