BIDAL-HCMUS@LSC2020: An Interactive Multimodal Lifelog Retrieval with Query-to-Sample Attention-based Search Engine

Anh-Vu Mai-Nguyen
University of Science, VNU-HCMC
Ho Chi Minh, Vietnam
1612904@student.hcmus.edu.vn

Trong-Dat Phan
University of Science, VNU-HCMC
Ho Chi Minh, Vietnam
1512102@student.hcmus.edu.vn

Anh-Khoa Vo
University of Science, VNU-HCMC
Ho Chi Minh, Vietnam
1512262@student.hcmus.edu.vn

Van-Luon Tran
University of Science, VNU-HCMC
Ho Chi Minh, Vietnam
1612362@student.hcmus.edu.vn

Minh-Son Dao
NICT
Tokyo, Japan
dao@nict.go.jp

Koji Zettsu
NICT
Tokyo, Japan
zettsu@nict.go.jp

ABSTRACT

In this paper, we introduce an interactive multimodal lifelog retrieval system with the search engine built by utilizing the attention mechanism. The algorithm upon which the system relies is constructed by applying two observations: (1) most of the images belonged to one event probably contain cues (e.g., objects) that relate to the content of queries. These cues contribute to the representative of the event, and (2) instances of one event can be associated with the content and context of such an event. Hence, when we can determine the seed (by leveraging the first observation), we can find all relevant instances (by utilizing the second observation). We also take a benefit of querying by samples (e.g., images) by converting text query to images using the attention-based mechanism. Thus, we can enrich and add more semantic meaning into the simple text query of users towards having more accurate results, as well as discovering hidden results that cannot reach by using only text queries. The system is designed for both novice and expert users with several filters to help users express their queries from general to particular descriptions and to polish their results.

CCS CONCEPTS

• **Computer systems organization** → **Client-server architectures**; **n-tier architectures**; • **Theory of computation** → *Unsupervised learning and clustering*; • **Information systems** → **Top-k retrieval in databases**; **Information extraction**; **Image search**; • **Human-centered computing** → **Web-based interaction**; • **Computing methodologies** → **Information extraction**; **Scene understanding**.

KEYWORDS

lifelog, clustering, image retrieval, feature extracting, content and context, semantic, attention mechanism

LSC '20, June 9, 2020, Dublin, Ireland

ACM ISBN 978-1-4503-7136-0/20/06…$15.00
https://doi.org/10.1145/3379172.3391722

ACM Reference Format:
Anh-Vu Mai-Nguyen, Trong-Dat Phan, Anh-Khoa Vo, Van-Luon Tran, Minh-Son Dao, and Koji Zettsu. 2020. BIDAL-HCMUS@LSC2020: An Interactive Multimodal Lifelog Retrieval with Query-to-Sample Attention-based Search Engine. In *Proceedings of the Third Annual Workshop on the Lifelog Search Challenge (LSC '20), June 9, 2020, Dublin, Ireland*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3379172.3391722

1 INTRODUCTION

Recently, lifelog analytics and retrieval has emerged as a hot topic in research communities. Lifelog is known as a dataset stored data captured by lifelogging that refers to activities of using personal devices (e.g., lifelog camera, smartphone, wearable sensor) to record a surrounding environment with the first-person perspective and personal data. With the target of supporting well-being life, many types of research have been conducted to get insights from lifelogs. One of the most exciting topics is Lifelog Moment Retrieval (LMRT), which tries to understand human activities from the first-person perspective. This topic can take further advantages in improving human memory, making people's qualification of living better both in social and physical activities. Being inspired by the interest and importance of doing in-depth research on this topic, research communities have established many events and challenges to create an environment for competing and sharing knowledge. Among those occasions, Lifelog Search Challenge (LSC) workshop makes much attractive because of its reality-based on providing an on-site challenge where participants have a limited time to answer given queries quickly.

The main challenge of LSC is to build a system that allows users to get relevant images in the shortest time but still be able to maintain the qualification of retrieved results. Another challenge of LSC is to design a system that can work for both novice and expert users. Novice users tend to use merely natural-language-liked queries, while expert users require more supporting tools to enhance their queries.

In this paper, we introduce a new interactive multimodal lifelog retrieval system with the query-to-sample attention-based search engine that is designed for the LMRT domain with the unique customization for LSC 2020 challenge [4]. We emphasize two significant contributions as follows:

- An attention-based framework flexibly matches words to their relevant images towards increasing the accuracy and

speed of searching, as well as minimizing the efforts users have to pay for handling the running flow.
- A friendly and convenient interactive system with supporting tools enhances users' queries, and polishes retrieved results towards having the optimal results that meet the desire of users.

The paper is organized as follows: section 2 reviews some related works, section 3 introduces our system, section 4 gives some examples of how the system works, and section 5 concludes our paper.

2 RELATED WORKS

Almost all systems attended LSC challenge have offline and online stages. The former aims to build a server containing enriched or well-structured data, which is useful for the online stage. The latter targets to provide users with an interactive interface to retrieve images

At the offline stage, a wide range of approaches is applied to preprocess raw data to form enriched and well-structured data stored in a server. In other words, at the offline stage, the data preprocessing plays a vital role in a retrieval system. One of the most common approaches to pre-process data is to enrich information extracted from images. To maximize useful information of objects and scenes extracted from images, single- or multi-modal schema is utilized to construct a detection/extraction model with various datasets. Regarding object detection models, COCO and Open Image V4 datasets are used to train Faster R-CNN with ResNet101 backbone [8], Yolo-v2, and Faster RCNN [2], and SNIPER network [9]. With respect to scene detection models, Place365 dataset is utilized to train ResNet152 [8], DenseNet[2], Place365CNN [9]. In order to enrich information extracted from images, different methods are conducted, such as removing blur images [2], creating tags from images' caption [12], building habit concepts [8], and constructing a keyframe of a bunk of images [10, 11]. In [6], the Exquisitor system applies a high-dimensional-feature-vector-based clustering approach to group similar images into semantic clusters. Datasets are also re-organized for better indexing and enhancing the speed of queries by using supports from MongoDB [9], Hash Table, and Tree [8].

At the online stage, most of the systems leverage web-based interfaces to allow users to find expected images in a variety of approaches. Firstly, there are two types of queries such systems allow users to use: textual- and visual-based queries. For textual-based queries, text queries are parsed into noun, verb, adjective components, and then their synonyms are found to form a bag of words. These bags of words are compared to a library of words stored in the dataset [8], ranked tags generated through Microsoft Vision API [12], similar concept words [2], labels and captions in the dataset [15], or processed with Free-text Ranking Algorithm [9]. For visual-based queries, similar-semantic-content images are used as the inputs of systems developed by China-team [12], VIRET [11], Le et al. [9], while sketched images are applied by lifeXplore [10] and Vitrivr [15]. Moreover, to take advantage of metadata, most systems construct a group of filters to provide users with the ability to form the new query whose content is the integration between the user's query and the metadata-based-search-keys from such filters, such as introduced in HCMUS-team [8], lifeXplore [10], VIRET [11], and Taiwan-team [2]. Enhancing the interactive search by suggesting users new results based on previously retrieved results selected by users is another approach in this direction, proposed in [6].

3 INTERACTIVE MULTIMODAL LIFELOG RETRIEVAL SYSTEM

The proposed method is built based on the following important observations:

- Each lifelog event usually has some special features that help to distinguish this event from others. Thus, one event can be queried by its characteristics that might be represented by some special and representative objects. For example, the representative object of the event "watching football on TV" could be a television object whose screen displays a green background and some moving players.
- 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).
- People can polish their search interactively when the previous rounds of results could give them more cues or hints to clarify their searching purpose (i.e., from abstract to detail query)

To realize the above discussion, our team design a system which contains two major parts:

(1) The query-to-sample attention-based search engine utilizes up-to-date deep learning techniques, clustering methods, and similarity search approaches for searching.
(2) The web-based user interface (WebUI) module allows users to manage their search (i.e., enrich multimodal queries, polish results), interactively.

Figure 2 shows that our integrated system includes UI interaction and deep learning core. Each sub-module (e.g., Query-to-Image, Clustering) in this figure will be described one-by-one in the rest of this section.

3.1 Overview

In this subsection, we briefly introduce our system architecture, as illustrated in Figure 1. When users submit their queries through the WebUI (i.e., Retrieval Application), this query will be sent to the Application Server. At here, the query preprocessing occurs, and the result is transferred to the Core Server. At the Core Server, the query-to-sample attention-based search engine is activated to search relevant images. The searched results included images and their IDs are returned to the Application Server to store in Cache and Storage, respectively. From the Application Server, the searched images are returned to the Retrieval Application and displays for

users. Images and their IDs stored in Storage and Cache are used for the interactive process.

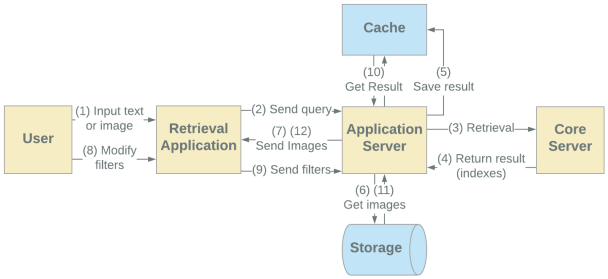

Figure 1: The System Architecture

3.2 Query-to-Sample Attention-based Search Engine

The proposed algorithm is described in Algorithm 1. First, we prepare our dataset by detecting all objects from images stored in the dataset. We extract features from these objects and stored in our database by utilizing *FAISS* [5], a library for efficient similarity search and clustering of dense vectors. We cluster images stored in our database into atomic clusters by using function *Cluster*, as described in 3.2.3.

When users express their query by text, we use the function *Attention* to create a sample whose content adapted with the original query to transform the query and the dataset to the same space. Such a sample may come from diverse sources, such as sketching and drawing, taking a photo, or the internet. In our experience, images from the internet are available and convenient to download for quickly generating the sample. We apply the attention mechanism, as described in 3.2.2, for generating samples from queries. Next, the function *ObjEmb* is activated to generate the visual features of the sample, namely sample-feature vector V_{sample}. This V_{sample} is used to query upon the FAISS-based database with the predefined similarity threshold. Then, the most similar feature vectors with V_{sample} are collected to form the new sample set. We loop this step until no new items are pushed into the sample set. The list of seeds is easily created by mapping items in the sample set (object vectors) with the corresponding original images. Finally, the set of seeds is used to find all clusters that contain these seeds. At this step, users can interactively manipulate and polish the results (e.g., select correct images manually, remove noisy images), as described in 4. Consequently, all lifelog moments satisfied with the query are found.

3.2.1 Feature extraction. We introduce the notation and all of the feature vectors which are used for our engine.

Notations and definitions.

- 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 A Attention-based Lifelog Moment Retrieval

Input: $Q, \{I_i\}_{i=1..N}$
Output: $LMTR$
{ONLINE}
if $type(Q) ==$ "$text$" **then**
  $Sample \Leftarrow Attention(Q, I)$
else if $type(Q) ==$ "$image$" **then**
  $Sample \Leftarrow Q$
end if
$V_{sample} \Leftarrow ObjEmb(Sample)$
{OFFLINE}
$\{C_m\} \Leftarrow Cluster(\{I_i\}_{i=1..N})$
$BoV \Leftarrow \emptyset$
$S \Leftarrow \emptyset$
$\forall i \in [1..N], BoV_{I_i} \Leftarrow ObjEmb(I_i)$
$BoV_{DB} \Leftarrow FAISS(BoV)$
$S^0 = S \Leftarrow S \cup \{V_{sample}\}$
while $S^i \neq S^{i+1}$ **do**
  $S \Leftarrow S \cup Query(S, BoV_{DB})$
  $i \Leftarrow i + 1$
end while
$seed \Leftarrow \{I_i | \forall V_{ik} \in S\}$
$LMTR \Leftarrow \{C_k | \forall j \in \|Seed\|, Seed_j \in \{C_k\}\}$
{ONLINE}
$LMRT \Leftarrow Interactive(LMRT)$
return $LMTR$

Details.

- We extract $\boldsymbol{V}_{o_i}$ from object detection model (Faster-RCNN backbone Resnet) in scaled Visual Genome dataset [7] (removing semantic overlapping classes)
- We extract $\boldsymbol{p}_i$ from place detection model described in 3.2.3
- We extract $\boldsymbol{V}_{w_i}$ via 2 steps. We extract hidden state 768-D vectors from BERT [3] are combined with one linear Conditional Random Field layer to construct seq2seq [17] model and output keywords (from a long input query sentence) with their representation vectors.

3.2.2 Attention-based Sample Retrieval. We introduce the novel attention mechanism approach for the Lifelog search engine inherited from the idea expressed in [1] that utilizes Top-Down Attention LSTM in a two-layer LSTM model for captioning images from feature vectors of regions detected by the object detection model [14]. Figure 3 illustrates how our method works to retrieval samples from a text query. Algorithm 2 depicts our algorithm. The following parts are a detailed explanation of our algorithm.

We observed the corollary about the attended image feature is either a convex combination of all input features or the vector representation for each word after post-processing the output vector of Language LSTM. As a consequence of a well-trained Bottom-up Attention model, we can find a useful feature transformation from word vector space to visual space.

For more specific, we reused the notation from [1] and our improvement to tackle the event searching problem. The input vectors

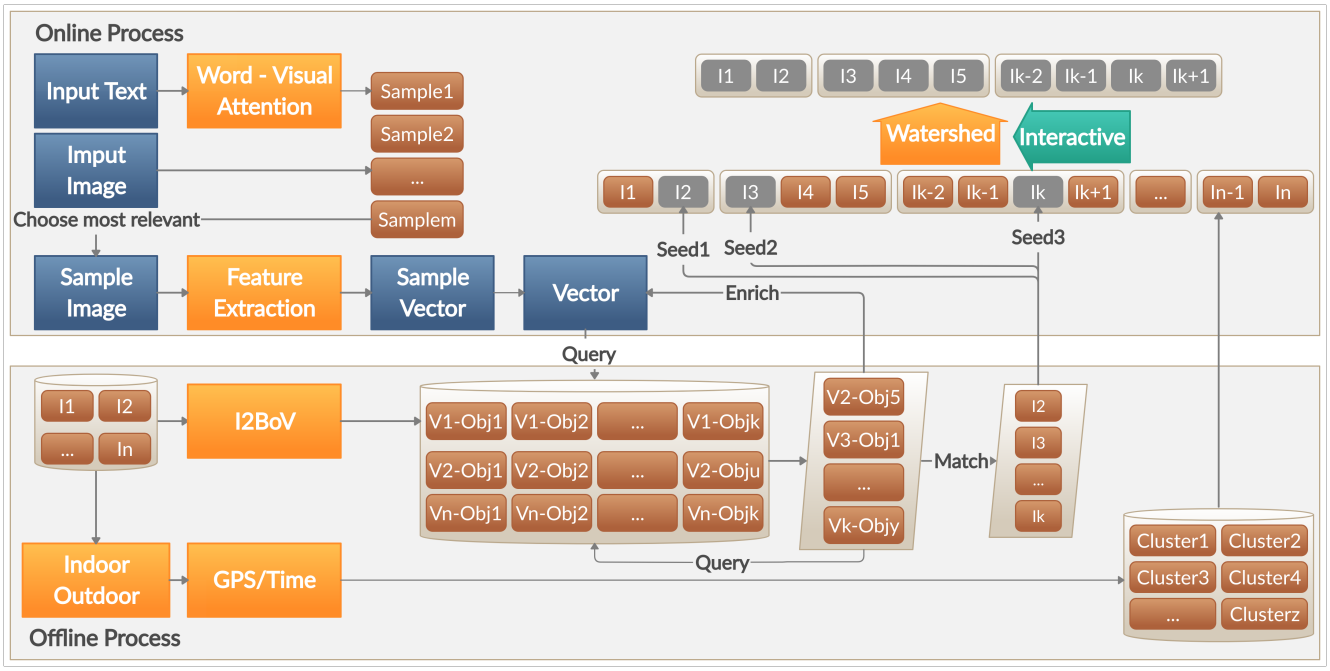

Figure 2: An Interactive Multimodal Lifelog Retrieval System

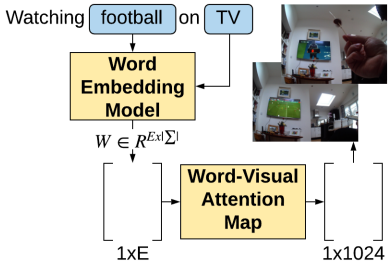

Figure 3: Attention-based samples retrieval

captioning model at each time step are given by:

$$x_t^1 = \left[h_{t-1}^2, \bar{v}, W_e \Pi_t\right] \tag{1}$$

where the Word Embedding $W_e \in \mathbb{R}^{E \times |\Sigma|}$ not only takes part of the input for the next step but is also used as the input space for our feature transformation. Moreover, the input to the language model LSTM contains the attended image feature, which is calculated:

$$\hat{v}_t = \sum_{i=1}^{K} \alpha_{i,t} v_i \tag{2}$$

In our method, we simplify the output for word-visual transformation which is also $\hat{v}_t = v_i$ where $i = argmax_{\alpha_{i,t}} \alpha_t$

Our approach is new and flexible for the lifelogging search engine due to the useful ability to transfer from word embedding space to visual embedding space. In many conventional methods [8] [12], the authors always search objects using labels or synonyms. Ironically, this is too hard and infeasible to work with natural language input. Therefore, our method of using such a word-visual transformation can retrieve images from text input using vector, which is much more natural without the human knowledge for choosing the most similar object label as before.

Algorithm 2 Attention-based Sample Retrieval Algorithm

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.

```
{V_wi}_{i=1..M} ⇐ WordEmb(Word)
{V_oj}_{j=1..N} ⇐ ObjEmb(Obj)
Training bottom-up attention model as in [1].
for all k ≤ ||V_w|| do
    v̂_k = Σ_{j=1}^{N} α_{k,j} v_j
    j' = arg max_j α_{kj}
    v̂_k ⇐ v_{j'}
    v̂_k is the optimized presentation for Word_k in visual space
end for
return {Word_i : Obj_j} where i = 1..M, j = 1..N
```

3.2.3 Clustering. Based on the second observation mentioned in , a bunch of clusters, called atomic clusters, is created by dividing the whole dataset into indivisible groups. We utilize the similarity of categories as the main criteria. We try to enhance low-quality category tags provided by organizers by using a semi-supervised learning method to regenerate new category tags for each image. First, we manually label categories for about 20k images. Then we

train a new model to automatically label the remained images in a dataset by utilizing the FixMatch method proposed by Google [16]. After completing the training, we run the image-by-image comparison by using generated tags to decide whether they share the same context and group similar images to one cluster. We put together the clustering method proposed in [13] with extracted vectors $\{\boldsymbol{p}_i\}$ as first mentioned in 3.2.1 to form our algorithm. Figure 4 shows some exceptional results generated by our work.

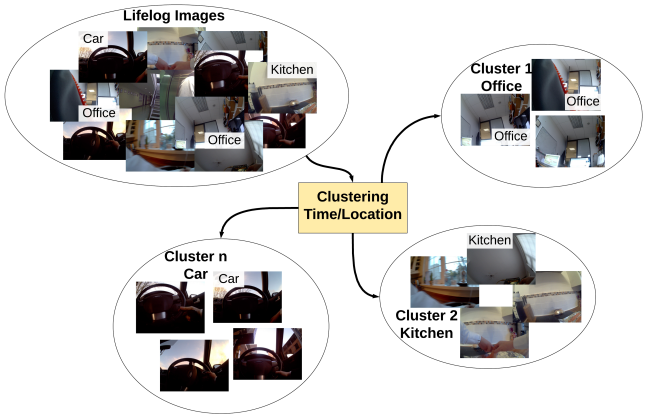

Figure 4: Atomic Clustering

3.2.4 Query. We utilize FAISS [5] as an engine to index visual feature vectors extracted as described in 3.2.1. We do query procedure as follows. First, we embed all vectors, using FAISS, into a unified database. Then we use the presentation of the sample retrieved from 3.2.2 (visual feature) as the replacement of the original input query (text). We use cosine similarity as the standard to compare sample vectors with indexed vectors. At the end of this procedure, all most relevant images are handled (via its indexes) by users.

3.3 Web-based user interfaces (Web UI)

Aiming to provide a friendly and convenient way to query data, we design our Web UI with the following functions:

3.3.1 Search bar. We allow users to create their queries with both texts and images (i.e., samples).

3.3.2 Filters. To enable the interactive retrieval schema, we provide users with a set of filters to let them polish their results as well as enrich their queries until they satisfy with the results. Figures 5 shows examples of using these filters to add more semantic cues to the query.

3.3.3 Results Visualization. We split the results visualization section into two lines. The above line shows the raw results after querying, while the below line contains inaccurate images removed by users. All images are sorted by time order. Users can arbitrarily move images from top line to bottom line and vice versa when clicking on them. With each image, we also display time information (e.g., day, day of the week, hour) and other semantic information. Figure 6 shows our description.

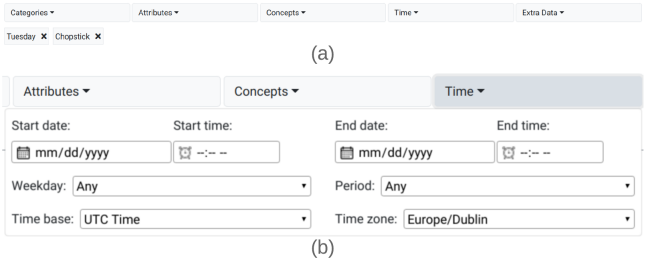

Figure 5: Filters

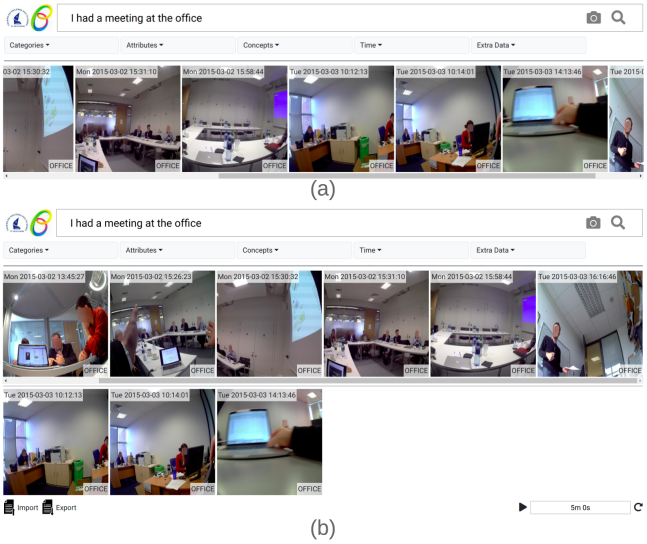

Figure 6: Results Polishing: (a) first result (b) polished result

3.3.4 Data Exportation. After filtering all not relevant items, users can use the exporting feature to get the result as a list of indexes. Users can also import the indexes into our system to visualize associated images.

4 INTERACTIVE PROCESS

In this section, we describe how the system works interactively (as illustrated in Figure 1), from the time a user enters his/her query to the time the final result is exported, as follows:

- Step 1: In the search bar, a user can both enter text query in and upload a relevant image. This task is handled at the Retrieval Application.
- Step 2a (Forward): the Application server converts the query into an acceptable format and sent such a query to the Core Server.
- Step 2b (Backward): the Application Server receives indexes of images relevant to the query from the Core Server, stores these indexes into the Cache temporarily, retrieves images from the Storage according to saved indexes, and returns those images to the Retrieval Application.

- Step 3 (Forward again): The user can modify the returned result by selecting the values of available filters. Then those values will be sent to the Application Server.
- Step 4: the Application Server receives the request, gets indexes from the Cache and applies filters to them, retrieves images based on filtered indexes, and returns those images to the Retrieval Application.
- Step 5: After receiving filtered images, the user can freely return to step 4 or go to step 6.
- Step 6: The user can manually remove unwanted images by clicking those on the first line or restore deleted images by clicking those on the last line.
- Step 7: If selected images meet the demand of the user, he/she can export the result to desired formats. If not, the user can return to step 6 or step 4.

To give an illustration of an interactive process, we introduce the example depicted by Figure 7. In this experience, we emphasize that the system can help users look back at their historical moments by expressing their requirements from general to specific queries. First, a user input the text query, "I had dinner at Asian restaurants." After the result is shown, the user wants to filter all moments that happened on "Tuesday". Then, he/she interests in the moments he/she used "chopsticks." Finally, the user can exactly know what he/she wants to look for: "find me all the moments I had dinner at Asian restaurants on Tuesday, and I was using chopsticks".

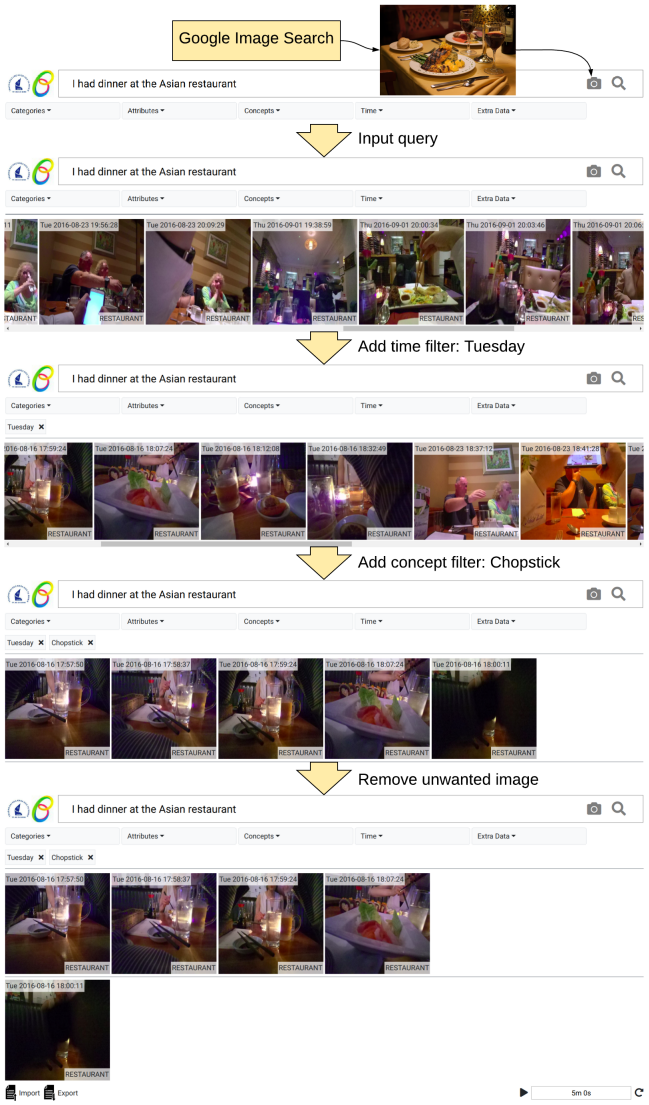

Figure 7: An example of interactive lifelog retrieval

5 CONCLUSIONS

We have introduced an interactive system for lifelog moment retrieval with the query-to-sample attention-based search engine. Our method tries to avoid a hard approach in understanding users' queries, soften the transformation process from input space (usually text) to latent space by applying attention mechanism. Furthermore, we also design a simple web-based interactive user interface to support query enrichment and output polishment. This WebUI not only prevents from getting typical troubles (e.g., wrong input's format) but also be a good supporter for verifying the correctness of the retrieval. In the future, we will compare our method to others, both those who attend LSC 2020 and those who work in the same domain.

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. We acknowledge the support of Science Foundation Ireland under grant number SFI/13/RC/2106 and L. Meltzers Høyskolefonds, UiB 2019/2259-NILSO.

REFERENCES

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. 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*. 6077–6086.

[2] Chia-Chun Chang, Min-Huan Fu, Hen-Hsen Huang, and Hsin-Hsi Chen. 2019. An Interactive Approach to Integrating External Textual Knowledge for Multimodal Lifelog Retrieval. In *Proceedings of the ACM Workshop on Lifelog Search Challenge*. 41–44.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018).

[4] Cathal Gurrin, Tu-Khiem Le, Van-Tu Ninh, Duc-Tien Dang-Nguyen, Björn Þór Jónsson, Jakub Lokoč, Wolfgang Hurst, Minh-Triet Tran, and Klaus Schoeffmann. 2020. An Introduction to the Third Annual Lifelog Search Challenge, LSC'20. In *ICMR '20, The 2020 International Conference on Multimedia Retrieval*. ACM, Dublin, Ireland.

[5] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *arXiv preprint arXiv:1702.08734* (2017).

[6] Omar Shahbaz Khan, Björn Þór Jónsson, Jan Zahálka, Stevan Rudinac, and Marcel Worring. 2019. Exquisitor at the lifelog search challenge 2019. In *Proceedings of the ACM Workshop on Lifelog Search Challenge*. 7–11.

[7] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. 2016. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. https://arxiv.org/abs/1602.07332

[8] Nguyen-Khang Le, Dieu-Hien Nguyen, Trung-Hieu Hoang, Thanh-An Nguyen, Thanh-Dat Truong, Duy-Tung Dinh, Quoc-An Luong, Viet-Khoa Vo-Ho, Vinh-Tiep Nguyen, and Minh-Triet Tran. 2019. Smart lifelog retrieval system with habit-based concepts and moment visualization. In *Proceedings of the ACM Workshop*

on Lifelog Search Challenge. 1–6.

[9] Tu-Khiem Le, Van-Tu Ninh, Duc-Tien Dang-Nguyen, Minh-Triet Tran, Liting Zhou, Pablo Redondo, Sinead Smyth, and Cathal Gurrin. 2019. LifeSeeker: Interactive Lifelog Search Engine at LSC 2019. In *Proceedings of the ACM Workshop on Lifelog Search Challenge*. 37–40.

[10] Andreas Leibetseder, Bernd Münzer, Manfred Jürgen Primus, Sabrina Kletz, Klaus Schoeffmann, Fabian Berns, and Christian Beecks. 2019. lifeXplore at the lifelog search challenge 2019. In *Proceedings of the ACM Workshop on Lifelog Search Challenge*. 13–17.

[11] Jakub Lokoč, Tomáš Souček, Premysl Čech, and Gregor Kovalčík. 2019. Enhanced VIRET tool for lifelog data. In *Proceedings of the ACM Workshop on Lifelog Search Challenge*. 25–26.

[12] Isadora Nguyen Van Khan, Pranita Shrestha, Min Zhang, Yiqun Liu, and Shaoping Ma. 2019. A Two-Level Lifelog Search Engine at the LSC 2019. In *Proceedings of the ACM Workshop on Lifelog Search Challenge*. 19–23.

[13] T. Phan, M. Dao, and K. Zettsu. 2019. An Interactive Watershed-Based Approach for Lifelog Moment Retrieval. In *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*. 282–286. https://doi.org/10.1109/BigMM.2019.00-10

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.

[15] Luca Rossetto, Ralph Gasser, Silvan Heller, Mahnaz Amiri Parian, and Heiko Schuldt. 2019. Retrieval of structured and unstructured data with vitrivr. In *Proceedings of the ACM Workshop on Lifelog Search Challenge*. 27–31.

[16] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. 2020. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. *arXiv preprint arXiv:2001.07685* (2020).

[17] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. 3104–3112.