Flexible Interactive Retrieval SysTem 2.0 for Visual Lifelog Exploration at LSC 2021

Nhat Hoang-Xuan[1,3†], Hoang-Phuc Trang-Trung[1,2,3†], Thanh-Cong Le[1,2,3*], Mai-Khiem Tran[1,2,3*], Van-Tu Ninh[4], Tu-Khiem Le[4], Cathal Gurrin[4] and Minh-Triet Tran[1,2,3*]

[1]University of Science, VNUHCM, Ho Chi Minh, Vietnam.
[2]John von Neumann Institute, VNUHCM, Ho Chi Minh, Vietnam.
[3]Viet Nam National University Ho Chi Minh City, Ho Chi Minh, Vietnam.
[4]Dublin City University, Dublin, Ireland.

*Corresponding author(s). E-mail(s): ltcong@selab.hcmus.edu.vn; tmkhiem@selab.hcmus.edu.vn; tmtriet@fit.hcmus.edu.vn;
Contributing authors: hxnhat@selab.hcmus.edu.vn; tthphuc@selab.hcmus.edu.vn; tu.ninhvan@adaptcentre.ie; tukhiem.le4@mail.dcu.ie; tmkhiem@selab.hcmus.edu.vn;
†These authors contributed equally to this work.

Abstract

With a huge collection of photos and video clips, it is essential to provide an efficient and easy-to-use system for users to retrieve moments of interest with a wide variation of query types. This motivates us to develop and upgrade our flexible interactive retrieval system for visual lifelog exploration. In this paper, we briefly introduce version 2 of our system with the following main features. Our system supports multiple modalities for interaction and query processing, including visual query by meta-data, text query and visual information matching based on a joint embedding model, scene clustering based on visual and location information, flexible temporal event navigation, and query expansion with visual examples. With the flexibility in system architecture, we expect that our system can easily integrate new modules to enhance its functionalities.

Keywords: lifelog, interactive retrieval, information system, joint embedding model, component integration

1 Introduction

People create a huge collection of photos and videos to capture daily activities as well as special moments in their lives. Such visual and related metadata are valuable sources for lifelog retrieval [1], not only to revive memories or to verify events but also to analyze for a better understanding of our behaviors and habits [2]. This can help re-design business processes, create better personalized intelligent services, improve personal lifestyle and health, etc.

Lifelog data consists of data in different formats such as photos, video clips, recorded audio clips, GPS information, personal healthcare data, data from different sensors, etc. To retrieve and analyze lifelog data efficiently, it is necessary to devise and develop a flexible interactive retrieval system that can support users to input their queries in different modalities. Therefore, the annual Lifelog Search Challenge (LSC) [1] has been organized to encourage different research groups worldwide to enhance their solutions for interactive lifelog retrieval systems.

Because of the wide variation in query types, we build a flexible query system that can integrate new query processing components to handle various query types. Our platform can also integrate different AI services to analyze lifelog data, e.g., object detection, scene classification and attribute detection, image captioning, etc. We also propose a mechanism to define service integration processes to deal with new scenarios for data analysis and query processing. In short, our Flexible Interactive Retrieval SysTem (FIRST version 2.0) provides five main features for lifelog moment retrieval as follows:

- Query by meta-data: Our system allows users to query with meta-data, including date, time, and location. We also extract scene text, entities, activities, and places to enrich meta-data for each photo/scene.
- Text query and visual information matching based on joint embedding model: Our system encodes both a text query and a photo into an embedding space to measure their similarity.
- Scene clustering based on visual and location information: The system visualizes scene clusters with two main clustering conditions: visual similarity and (GPS) location.
- Flexible temporal event navigation: Users can navigate visual lifelog photos with a flexible timeline to shrink or expand the time interval of interest.
- Query expansion with visual examples: From one or several example photos, our system can retrieve and rank photos based on their similarity and dissimilarity.

The content of this paper is organised as follows. In Section 2, we briefly review related approaches for lifelogging retrieval. We introduce an overview and main components of our system in Section 3. We then illustrate some usage scenarios of our system in Section 4. The conclusion and discussion for future work are in Section 7.

2 Related Work

Lifelog retrieval systems are user-centric, therefore there is an emphasis on ease of interaction and effectiveness of querying methods. In the past years, systems for LSC have evolved, integrating advanced features to help experts quickly navigate through the dataset. Each system has their own user interface (UI), some even have a different mode of interaction (e.g., virtual reality and voice input). We review the past systems and highlight their traits below.

The most traditional interface is arguably the textual search bar, which is the case for most popular search engines like Google Search. Due to its familiarity, the adoption of such interface makes it easier for novice user to grasp. This approach is used by the long-standing system vitrivr [3], as well as FIRST 2.0 [4], LifeSeeker 3.0 [5], and various other systems. These systems fill most of their UI with the list of retrieved images, which allows fast exhaustive browsing. With a good semantic representation in their index backed by state-of-the-art models, they generally achieve good results in the LSC while at the same time being easily accessible to users.

For practical purposes, some authors propose to put more emphasis on the visualization of the data, rather than relying on implicit semantics. Examples include [6] and [7], both trying to embed the data points to a three-dimensional space, which is natural to human. The later also makes use of virtual reality in an attempt to achieve an immersive experience. This approach allows user to have an overview of their data, for instance by recognizing clusters. While intuitive and interactive, these systems still rely on generated tags, which can also be efficiently indexed and utilized by traditional systems.

On the other end of the spectrum, systems augment the traditional search bar with tools such as canvas queries, or temporal expansion, in the case of SOMHunter [8] and Myscéal [9], respectively. Since they are feature-rich, when operated by system designers, they are among the best performing systems in the LSC competition. However, their complexity means novice users require more time to familiarize before they can operate the tool at its maximum potential.

In an attempt to strike balance, some authors [4, 10, 11] propose to use multiple views or overlay windows to keep the main UI clean. This approach enhances modularity and allows flexible integration of components, which is aligned with our design choices.

To summarize, while the inclusion of more features give users more tools to work with, they have to be accommodated by the UI and hence can make it clumsy and crowded, if not well-designed. Moreover, users have to be accustomed to them to search effectively. We choose to provide users with the core functions that we deemed important, and instead focuses on the user experience, specifically by devising a strategy so users can maximize the potential of our tool. We give more details on this in later sections of this paper.

3 Lifelog Event Retrieval with Flexible Components

3.1 System Overview

Since version 1.0 [12], we have designed our system to be flexible and hence can support the integration of different components. The system employed in this paper is based on FIRST 2.0 [4], and the specific components utilized in that version can be found in the paper. For this work, we mainly focus on a selection of vital components, that we have observed from our experience, are often used and play an important role in the retrieval process.

In the LSC competition, images are accompanied by additional metadata such as GPS, time, visual concepts, etc. This is valuable information as they can be indexed easily and assist greatly in the searching process. Therefore, we provide options for the user to leverage this information. Specifically, in Section 3.2, we show concepts can be defined and used in tag-based filtering. Section 3.3 describes the time filter functionalities that the user can use to narrow down the time range to search for. These modules can be combined in form of nested filters.

Apart from metadata, we also support visual similarity searching, in case the concept is not present but an example is. This module is described in Section 3.4. Finally, since lifelog data is ordered in time, the user may wish to browse through the collection, or expand forward/backward from a specific point in time. This is possible with our View Timeline feature, which is outlined in Section 3.5.

While having an arsenal of tools is preferable in some cases, in Section 6, we propose a strategy to effectively make use of the available tools to handle the queries. We demonstrate that this strategy translate to effective retrieval by novice users in Section 4, even without the presence of an OCR system.

Figure 1 shows the overview structure of our proposed system. We use the user-interface and system integration platforms from our FIRST version 1 [13] to manage different layouts and user interaction modalities, and query processing components, respectively.

3.2 Query by Concepts

Many search engines use tags in addition to the main query as an enhancement. In the LSC, this is no exception; some systems solely rely on them [7, 11]. For FIRST, as our main query is processed with a text-image joint-embedding model, we consider the use of tags as a complement, in the sense that tags are explicit and intuitive, while embeddings are implicit and model-dependent.

The list of tags to be included is pre-defined based on how the tool will be utilized. For LSC, the existing metadata is utilized first. Of the provided metadata, the location information is especially important, as it is occasionally directly mentioned in the query. FIRST 2.0 [4] explicitly support location tags,

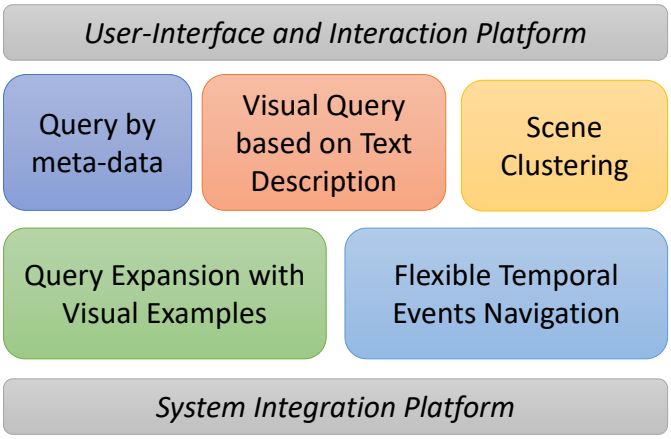

Fig. 1: System overview of our query system.

and also enriches the data with object annotations from pre-trained object detectors.

FIRST 2.0 [4] also introduced the idea of *hierarchical relationships*, for example in the case of a specific location (*café*) within another broader location (*city*). This helps the user to gradually narrow down the search, with the potential to backtrack in case a dead end is reached. The idea is not limited to just locations, and can be generalized to any class of concepts. In this work, we elect to only use the location tags to reduce the number of tags, as the semantic concepts are well-represented using our embedding model.

3.3 Filtering by Time

Human activities are closely associated with time, from long-term ones such as profession, to everyday activities including eating, meeting, etc. To recall a certain event, people can try to locate the time period, along with other adjacent events. Therefore, it is crucial that the user can choose a frame of time to search from. As the user browses events from that time period, they will remember more precisely, gradually shrinking the time period to a single point. This strategy is discussed further in Section 6.1.

The feature is implemented as a faceted search. Typically, time information will be provided gradually in the query, and the last hint can be as specific as a single date, therefore all date units (day, month, year, weekday) are present. On the other hand, the time unit (hour, minute) can be used to specify the parts of day (morning, afternoon, etc.) or to define the time where certain events are more likely to happen (e.g., a meal around lunch/dinner hour).

3.4 Query by Visual Similarity

From the information retrieval perspective, it can be difficult to index an image, due to its nature of not being made of smaller "pieces". However, with the application of deep visual models, images can be encoded into a vector, and the vector space model can be used to find the degree of relevance between images. For example, by taking the features of an image classifier, a pair of image with small distance is likely to contain the same concept. This high-level semantic representation makes it possible to generalize query expansion techniques to images, particularly by searching for images that are similar to a given one.

As the descriptions are generally object-centric, FIRST 2.0 [4] uses ResNet152 features to represent an image. The trend of using object detectors features is also seen in other systems [6, 9]. For FIRST 3.0 [14], the visual features are greatly enhanced and used to search with different level of details in an image.

3.5 Temporal Navigation

Besides photos or video clips, lifelog data may contain additional data from different sources, e.g. GPS, location, time, personal health data, etc. Such data is valuable to narrow down the search space for the moments of interest. We support the hierarchical relationships between locations, such as *Helix Cafe* in *Dublin*, and *Dublin* is in *Ireland*, etc. We also support searching based on date and time range, or day in week. Figure 2 presents the form to query based on location and time.

In our system, we further analyze to extract enrich meta-data with semantic concepts from photos. We use scene text [15] to augment information for objects and places, such as the brand name of a product, a number appearing in the scene, or the name of a place.

We extract both general and personalized objects [16, 17]. General entities can be detected with existing pre-trained object detectors, such as Faster RCNN[18] or EfficientDet[19]. We also train our own object detectors to localize items that are unknown to generic object detectors but usually appear in personal activities, such as *coffee machine* or *medicine cupboard*.

3.6 Text Query and Visual Information Matching based on Joint Embedding Model

Text description is one of the most effective ways to express the context of an image. Being able to input natural text description will enhance the capability of lifelog retrieval system and allow us to modify the query description freely based on our domain knowledge. Therefore, we create and integrate a multi-modal retrieval model called Self-Attention based Joint Embedding Model (SAJEM [20]) in our retrieval system FIRST version 2. As illustrated in Figure 3, the model contains two branches to process visual and text data.

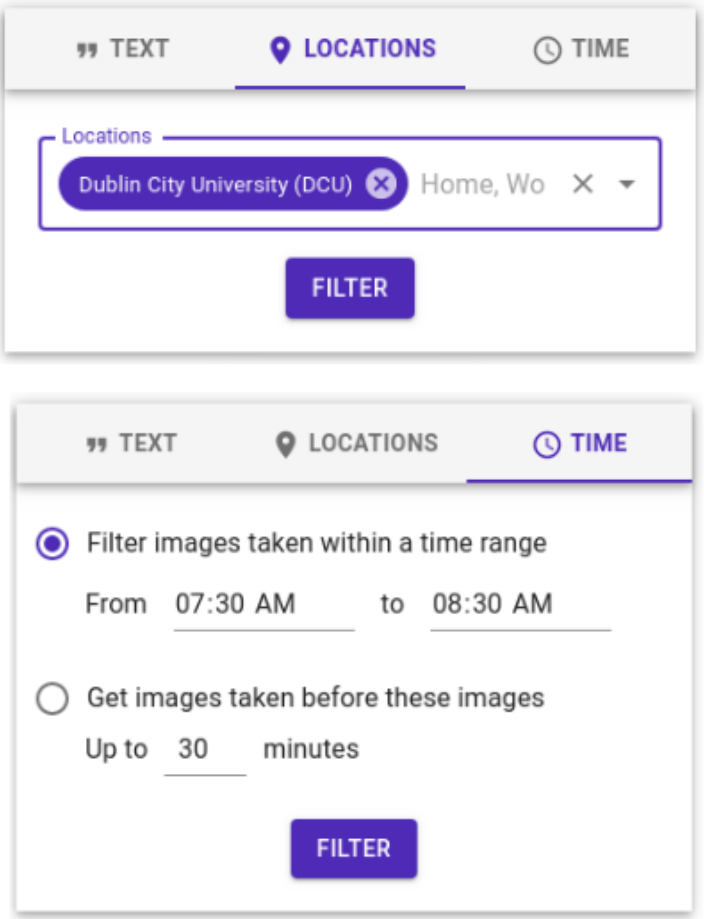

Fig. 2: Query by location and time.

In the image branch, we first use Bottom-Up Attention Faster-RCNN model [21] to extract the region-level features of images. The model is trained on Visual Genome dataset [22] so the extracted feature will have information about type and attribute of object. Each image now can be represented by 15 region-level features with the highest confidence scores. The next module of this branch is Self-Attention Module, which helps to learn the interactions (or co-existence) between objects in the image. We adopt the idea of Multi-head Self-Attention module from Transformer model [23] with some modifications: use dot-product attention instead of scaled dot-product and only use one head to calculate query, key and value vectors. After attention step, we apply Average Pooling on these vectors to form an unified vector for the whole image. Each image is now represented by a 2048-dim vector.

In the text branch, we need a model that can effectively learn the context of a sentence. With the rise of Transformer-like models since it came out, they became the standard for understanding textual meaning. Therefore, we use RoBERTa model [24] to encode the text sentence. We concatenate the [CLS] token vectors of the last 4 layers instead of just using the token of the last layers to capture more information. Each description will be represented by a 3072-dim vector.

After constructing one feature vector to represent for each domain, we use two simple Multi-layer Perceptron to map those vectors into a joint space, which has 2048 dimensions. Now we can easily calculate the similarity between an image I and a caption C through cosine similarity:

$$S(y_I, y_C) = \frac{y_I \cdot y_C}{\|y_I\| \|y_C\|}$$

with y_I and y_C are the final features of image and caption in joint embedding space.

To train the model, we adopt the Margin Ranking Loss from [25], which penalizes the model according to negative samples:

$$L(y_I, y_C) = \max(0, \alpha + S(y_I, y_C^-) - S(y_I, y_C)) + \max(0, \alpha + S(y_I^-, y_C) - S(y_I, y_C))$$

where α is non-negative margin constant. y_I^- and y_C^- are hardest negatives in the current mini-batch for y_C and y_I, respectively.

We train our model on MS COCO dataset [26]. This dataset contains 123,287 images and 616,767 captions so each image has roughly 5 captions on average. We keep the object detector fixed and train all other components for 10 epochs on COCO training set. We achieve the Recall@10 of 0.742 on COCO 5K Text-to-Image task, which is comparable to other text-image matching models.

3.7 Scene Clustering based on Visual and Location Information

Clustering images help users to handle images in groups easily. As mentioned in [13], our system aims to support different criteria for image clustering. Currently, we implement two common strategies: grouping images based on their visual similarity and location.

Figure 4 demonstrates the visualization of image clusters based on their visual similarities (with visual features extracted by ResNet152). For photo clusters based on geolocation, we visualize them on map so that users can perform query based on locations.

3.8 Flexible Temporal Events Navigation

To find or verify a certain event, we may need to look backward or forward a starting time instant. This idea was proposed and implemented in our previous

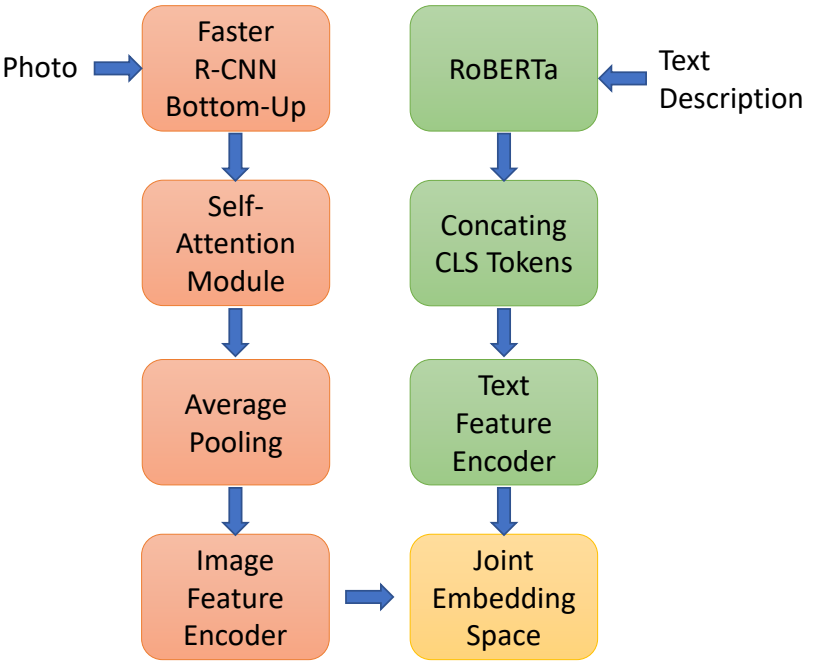

Fig. 3: Main steps for Joint Embedding Model for visual and text feature comparison.

Fig. 4: Visualization of image clusters[13].

systems, such as LifeSeeker 2.0[27] or FIRST 1.0 [13]. From a seed point, an initial photo corresponding to some criteria in the query, we can expand the

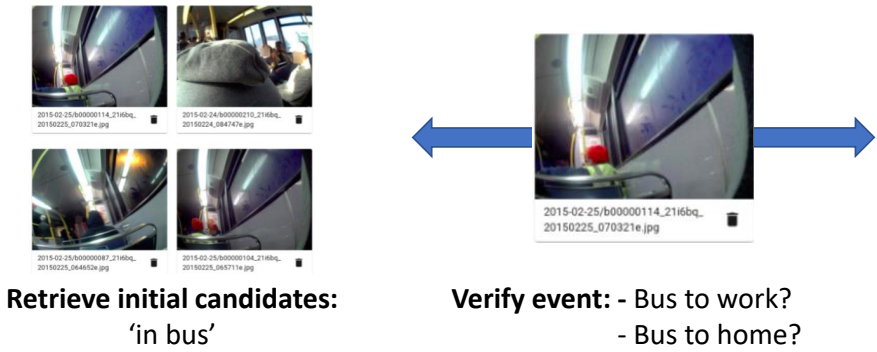

Fig. 5: Example of sequence navigation for event verification.

sequence of photos to freely navigate in the timeline to refine the result, or to check for another event happening before or after the initial event.

As users may need to navigate slowly or quickly across time from a photo, our system supports users to adjust the time step interval, the granularity of the temporal data, to visualize the photos at specified intervals using the time slider. The step interval varies from seconds to hours, even across days, depending on the situation.

Figure 5 illustrates an example of sequence navigation for event verification. We can easily retrieve candidate photos corresponding to *in bus* environment, then expand the sequence of photos surrounding an initial photo to check the destination of that bus trip.

3.9 Query Expansion with Visual Examples

To query similar images, we use ResNet152 features extracted from an initial photo and other photos in the dataset. Each photo is represented as a 2048-dimension feature. To speed up the process, we pre-calculate and store visual features of all photos in the collection. To evaluate the similarity between photos, we use the cosine distance between their features.

3.10 Service Integration for Lifelog Data Analysis

Because lifelog data has a large volume of data and needs many different processing tasks on it, we propose and build a solution using Google Colab and flexibly integrate Lifelog data analysis services. With built-in service integration, our system allows to change or add new processing features quickly by replacing and assembling processing modules through exporting results from API services. As needs change, systems and services can change according to needs.

Google Colab is a suitable choice for us to build data processing and analysis services. Because it provides servers with integrated GPUs for powerful

modeling capabilities and rapid compatibility with cloud storage like Google Drive to leverage storage, security, and permission.

Various lifelog data processing services such as object detection, place detection, image captioning, etc., can be implemented and deployed on different Google Colab instances as different servers. Each serving instance handles a particular analysis and exports the results to the outside via the API. Services all share the same process, including pre-processing, main processing, and post-processing in Figure 6. Pre-processing is responsible for receiving input from the API requests then normalizing the input for analysis. Main processing is the main processing function that uses algorithms and models depending on user needs. And post-processing is the part that takes care of saving the processing results to cloud storage. The analytics services are then treated as separate APIs on instances of Google Colab. When the system needs to process lifelog data according to one or more different types of analysis such as object detection, place detection, image captioning, etc., it only needs to call the right APIs.

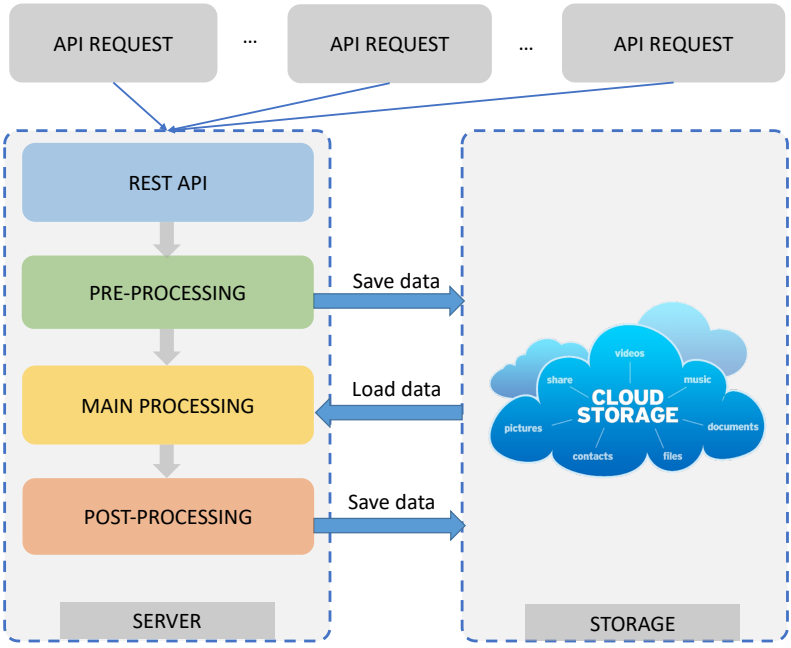

Fig. 6: Flexible data analysis service for lifelog data.

3.11 Forming a Pipeline by Linking Components together Using NodeRed

Due to the nature of the solution where data is processed in a diverse, highly componentized environment, we propose a method to utilize existing technologies to accelerate the development process, while maintaining the final quality. Without having to re-implement existing functionalities, more time can be spent into research and development for critical components. Furthermore, existing tools are independently developed and thus, its stability is guaranteed. NodeRED is chosen for this purpose, because it provides a decent platform to design data flows, as well as orchestrating the execution.

On the front-end, NodeRED's user interface is intuitive for even users without a technical background. NodeRED uses the blackbox approach, where each component that represents a function are represented as nodes on the design surface. Each nodes have its own sets of input and output where data is received and sent, respectively. Nodes of functionality are laid out by dragging and dropping from the available nodes. The flow from the source to destination can be made by connecting inputs and outputs of nodes together. This therefore, establishes an execution graph that is easily understood by both humans and machines.

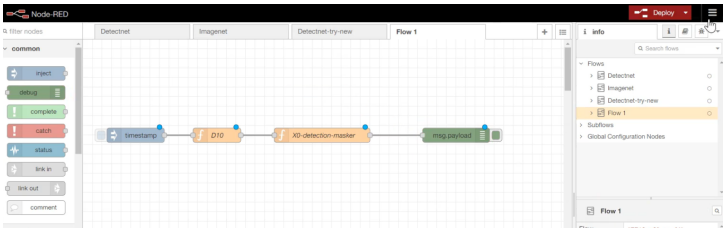

Fig. 7: NodeRED design surface, with nodes connected to each other.

On the back-end, NodeRED uses NodeJS, a popular open-source, cross-platform, JavaScript runtime environment. Therefore, creating a new node for NodeRED is simple and standardized as creating a module for NodeJS, which opens the possibilities for automated updates and programmability.

4 Some Usage Scenarios

In this section, we demonstrate the capacity of our system by showing some scenarios and how we can apply the features described in Section 3 in a practical situation. While the developers have a good understanding of the features, novice users might not have the same experience; this difference is further examined in Section 5. To explain the philosophy behind the feature used in each situation, Section 6 provides some guiding principles on how to best use our system.

Scenario 1: *Following someone's red backpack after having a coffee in Angelina's Cafe on a cold day...*

We start by filtering images at *Angelina's*. This step yields a few dozens of images, but notable they were all taken in a single day. Taking advantage of this information, we use the Timeline View feature to quickly look for moments after having coffee and easily locate the red backpack. Figure 8 shows this process.

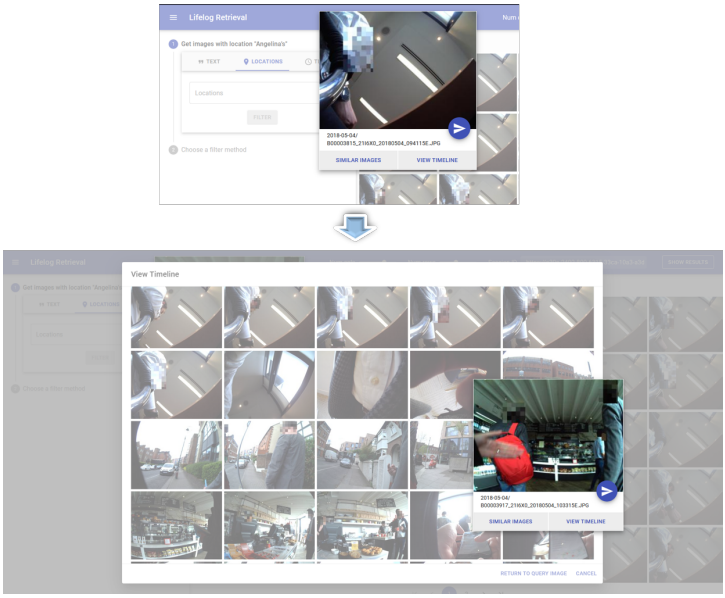

Fig. 8: Filtering images at *Angelina's*, followed by timeline navigation.

Note that this image can be found without using any text query. While this might be unintuitive for novice users, it is completely normal to successfully solve a query in this manner.

Scenario 2: *Looking at ancient Chinese vases in a museum. There were two of them. They were blue and white in a wood and glass case.*

For this query, we search for *blue vases in museum* and find a very promising candidate as the top-1, shown in Figure 9. It is a good practice to confirm the result using the given information. In this case we look at the photos taken before and after that moment to confirm the location to be a museum.

Scenario 3: *Someone was looking inside the medicine cabinet in the bathroom at home.*

We use the query (in text) "looking inside the medicine cabinet in the bathroom" to find some initial result, then we find similar images (query expansion) with ResNet152 feature to find remaining images. The query expansion function is useful to search for all events corresponding to a query (Figure 10).

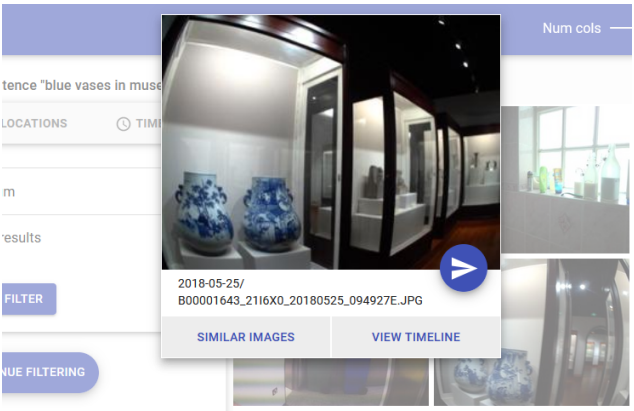

Fig. 9: Searching for *blue vases in museum.*

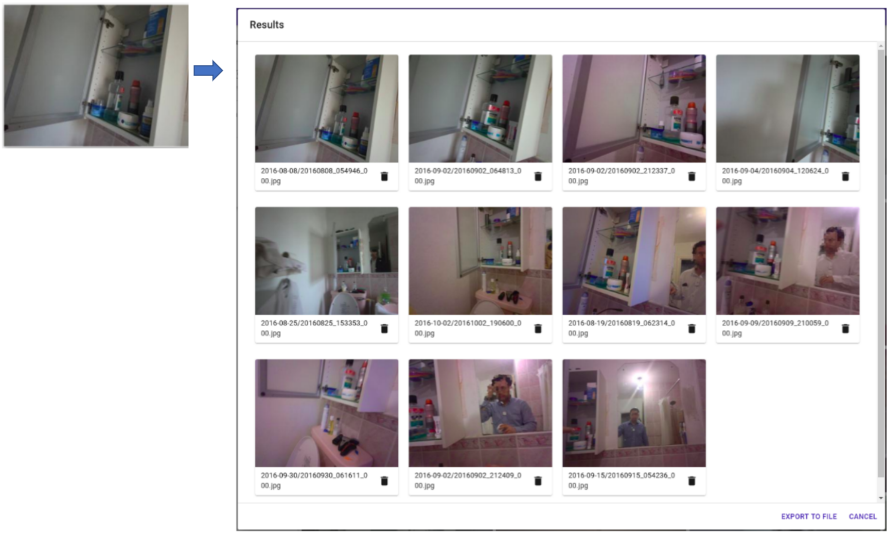

Fig. 10: Use query expansion to find all events corresponding to a query.

Scenario 4: *A woman in red top standing in front of a poster.*

We find one image when searching with the query (in text) *a woman in red top standing in front of a poster*. To find another result, we perform the two-step search. First we retrieve top 1000 images with the query *a woman in red*, and refine the result with another query *poster on the wall*. Figure 11 shows the two photos corresponding to this query.

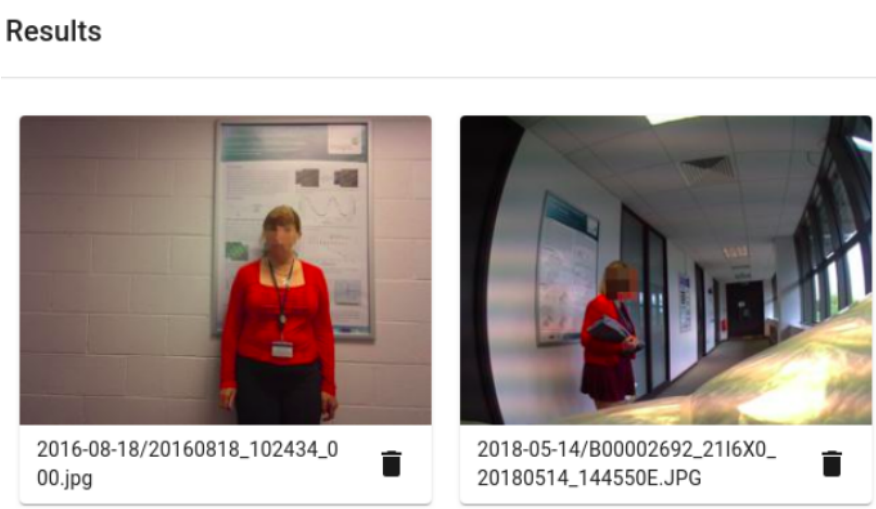

Fig. 11: Iterative query with multiple steps.

5 User study

From the first years, the LSC has accommodated a novice session to ensure the systems are actually operable by other people. However, the definition of "novice user" was not clearly determined, it is mostly understood as "not the system developer". We note that this definition covers a wide spectrum of users, which have different skills. This can lead to varying novice user evaluation performance, which makes it difficult to assess the systems. In this work, we conduct a user study to analyze in detail the behaviour of the novice user, and point out the differences between them and experts. This approach is different as it is centered around the user itself, and it seeks to provide more insights about the retrieval process itself. This information is used to devise the best practices described in Section 6, which could bridge the gap between the novice user and the experts.

5.1 Background

There is a known discrepancy in performance between experts and novice users [9, 28]. However, as noted before, there is no clear definition of a "novice user". In practice, in LSC'19, novice users were system developers from other teams, while in LSC'20, they were conference attendees. We can already see there is a disparity in skills between the two groups and other users, for example volunteers recruited at an university. To better quantify this difference, we propose to look at the differences in two areas:

- **System familiarity.** This is the ability to fluently operate the system. It is expressed through the ability to quickly and effectively locate and

use the correct tools to perform the user's *existing intentions*. This can be gained by spending time using the system or through training.

- **Searching proficiency.** This describes the user's capability to perform searching tasks. It concerns whether the user knows which strategies one can apply while searching, which term to search for, the specifics of lifelogging data, etc. One can improve by adapting to the competition and the field in general.

With this distinction, we see that the current perception of "novice user" is somewhat skewed towards inexperience in the first area, where existing literature [9, 11] explore and focus on usability improvements. Note that system developers can simply learn to adapt to the features of another system based on their own system and experience, while observers at the conference generally have interest in the competition and have an idea of the formulation of the tasks, therefore they both have some expertise in the second area. We propose to broaden the current view by looking at the effects of searching proficiency, where people with little familiarity to LSC topics cannot effectively search, irrespective of the system used.

5.2 Setting

We conduct our study on 5 volunteers who were not familiar with our system nor the competition. They were first thoroughly introduced to the system and its features and then practiced on 3 test queries to gain *system familiarity*. Because the system is quite similar to ordinary search engines and does not contain complex UI elements, this process does not take much time. The setting is the same as LSC, with each query consisting of 5 incremental hints, each hint lasting for a minute, therefore the total time for a query is 6 minutes. Furthermore, all of the queries used were from LSC'21 itself. The list of queries is as follows:

Q1 I was taking a photo of a lake with a DSLR camera. It was my Sony camera. I was driving outside of Sheffield before and after stopping at the lake. It was in 2015 on a Saturday.

Q2 There was a Blue Air aircraft... after I got off a flight. I was walking across the airport ground before going through the airport building and getting a taxi. I had just arrived to Dublin in May 2018.

Q3 Planning a thesis/dissertation on a whiteboard with my PhD student, who was wearing a blue and black stripey top... in my office in 2016. We were using blue, black and green pens. After this I went back to work at my computer. It was on the 27th September.

Q4 I was lost and looking for directions on a street, close to an asian restaurant called Maple Leaf. It was in the late afternoon or evening and it was in Wexford. I had driven there in 2015.

Q5 I remember the TagHeuer advertisement for a watch. It was a footballer and a watch. The footballer was sideways kicking the ball. It was right before I went down stairs in the airport in Frankfurt... in 2015.

Query	U1	U2	U3	U4	U5
Q1					
Q2	78.75	94.58	80.14	79.03	
Q3		50	93.75		80.83
Q4		98.89			
Q5					

Table 1: The scores of the 5 novice users in the study. Blank cell implies a score of 0 (i.e., the target image is not found).

A total of 5 queries were used, with the result depicted in Table 1. The queries were not ordered in any particular way. Note that the system experts were able to solve all of these topics in LSC'21, so they can compare various searching strategies for them.

5.3 Novice and expert strategy analysis

We look at the novice users' approaches to the queries and compare them to the experts'. Leibetseder and Schoeffmann [11] previously provided a perspective on different strategies between novices and experts for a particular query. We seek to find a more detailed explanation through analysing multiple queries, and from that derive the best practices for all users in Section 6.

Table 2 shows the detailed strategy of the novice users and the corresponding strategy for the expert user. For clarity and conciseness, we only show and analyze a subset of the strategies for some of the queries.

Query	Expert strategy	Novice strategy
Q1	"2015" → "water"	"DSLR photo of a lake" "photo of lake" → "2015"
Q2	"blue aircraft after got off a flight"	"blue air aircraft"
Q3	"text on whiteboard"	"whiteboard and people" "planning a thesis on a whiteboard" "whiteboard phd student"
Q4	"restaurant on street"	"Maple Leaf restaurant" "asian restaurant" "on a street"
Q5	"Frankfurt Airport" → view timeline	"Frankfurt Airport" → "footballer" "down stairs in the airport" → "Frankfurt Airport"

Table 2: The detailed strategies of novices in comparison to the expert. Text in quote denotes a search operation, while underlined text indicates a filter. "view timeline" is the operation in Section 6.4. Each line describe an attempt, green denotes successful strategies (i.e., the user finds the correct image with that strategy), while red denotes failed attempt.

In queries ***Q1*** and ***Q3***, we see a common problem with novice users: they usually rely too much on the prompt, often inputting the whole prompt as-is. The original prompt often contains information that is unnecessary ("PhD

student" vs. "student"), or not understandable by the system ("DSLR photo", "planning a thesis"). This redundant information usually harms the quality of the returned result, hampering the search process.

In ***Q4***, the expert elected to combine two concepts (restaurant and street) and thus was able to find the result relatively quickly. None of the novice users were able to achieve this, they either focus on looking for a restaurant with the specific name, or try to find a restaurant from street images. The only successful novice strategy was possible due to some modest OCR capability of the joint-embedding model, however even in that case the expert strategy still resulted in a higher ranking (top-5 vs top-9).

For ***Q1*** and ***Q5***, the importance of metadata information (time and location) is highlighted, as they greatly aid in the search of seemingly too general queries. In the actual competition, these clues are often left to the last minutes; there is a phenomenon that teams will submit very quickly when these critical clues are revealed. However, this is not the case for novice users; we saw that the addition of information usually brings confusion, evident through the fact that there is almost no submission with low score (i.e., no submission near the end of time).

To solve ***Q5***, both expert and novice users have to wait till the location clue to perform meaningful searches. This clue reduces the search space to a single day, which the expert utilized the view timeline feature to look ahead in time and find the correct moment. The novices were not able to make use of this information, however. They still resorted to the familiar text-based search even when the search space is very small.

5.4 Results analysis

After observing, we noticed a few shared imperfections that is usually absent from experts, those were:

- **Imprecise and limited query formulations**. Novice users tend to have an over-reliance on the provided prompt, which might be not concise enough. Expert users, in contrast, focus on few concepts and their interaction to form a succinct query. They are also more flexible in changing the description or choosing alternative words, in case the original one is ineffective.
- **Ineffective handling of abundant information**. Experts are quick to react to a critical clue, evident through the prevalence of last-clue submission in the actual competition. In other words, experts perform nearly optimally theoretically, being able to quickly answer when the searching space is significantly narrowed. However, novice users have not reached this level, they can get confused when presented with extra information. The reason is twofold: first, they are not aware of the theoretical effectiveness of filters, e.g., knowledge of the month is on average a 12-fold reduction. Second, novice users have not utilized multiple/nested filters to their best.

- **Underuse of visual search and view timeline functionalities**. Apart from text searching, other visual explorative functionalities are also built-in many systems to enable fast searching in the visual domain. However, it has to be emphasized that since everyday usage of web search engines does not concern these features, novice users are not expected to take them to account from the get-go.

In conclusion, while time pressure and unfamiliarity definitely play a role, we believe that it is the user's overall approach to retrieval that matters the most. Therefore, in Section 6, we suggest some principles that tackle these shortcomings in user's approach, with the aim to elevate the user's search effectiveness by giving them more ideas to work with.

6 Best practices

As stated in Section 5, there is a fundamental difference in how novice users perform their search, which limits what the novice user can do. To help new users maximize the potential of our tool, in this section, we provide a few guidelines based on our experience. They are not concrete steps, rather, they serve as "hints" on which tools the user may use in a certain situation. The end result still depends on the creativity and flexibility of the user in utilizing the provided tools.

In the following discussion, we use this example query to demonstrate the strategies that a user can consider. The answer for this query is in Figure 12.

Q: I was having lunch with a colleague. It was in 2016. I remember it was at a Subway, I bought a burger and a bottle of water. The day was 13^{th} September.

Fig. 12: The answer for the example query

6.1 Apply filters, ordered by certainty

Filters are great data indexer and have great practical usability, as discussed in Section 5. Therefore, they should be used whenever possible. As its core, the FIRST system uses a nested filtering pipeline. In other words, the result of the first filter is used as the input for the second filter, and so on. Therefore, the order of the filters are important.

For the above reasons, we recommend that filters are applied in **decreasing order of certainty**, i.e., conditions that are definitive should be applied first. Some examples include *time* (month, year, weekday), *location* (work, home), *objects* appearing in the prompt (bottle, burger, etc.). The benefits are two-fold: first, they quickly reduce the search space, making it more manageable for both the human operator (less results to browse through) and the system (less processing). Second, as the less certain filters are applied later, they can be changed without greatly affecting the established direction of the search. In the example query, a common approach is to start with filtering the year (*2016*), then following with a customary text query (e.g., *having lunch with a colleague*). Since the text query succeeds the metadata filter, it can be conveniently modified.

6.2 Vary the search term

The single textual query input method is simple, yet flexible in the sense that the user can formulate the search terms in different ways, owing to the joint embedding model described in Section 3.6. The user can choose to search for objectual concepts or more abstract ones, such as human activities, human emotions, events, etc. There is no clear distinction on which way is better, their viability depend on the specific circumstances.

To maximize the chance of success, we recommend to choose the description that provides the most **specificity**. In other words, the user should favour the concept that is most unique to the sought scene. For the example query, *having lunch* is a good start, yet there are expectedly too many relevant results for that term. On the other hand, *burger and water bottle* is a much more specific one and works well enough in this case. Additionally, when the current approach does not work, the user should not be hesitant in changing the chosen concept, because with correct ordering explained in Section 6.1, it should be easy to change the query.

6.3 Use visual search

The visual similarity search is another functionality that gives a great deal of variability in searching methods. It is especially helpful as a query expansion mechanism when the concept cannot be well described with words, or in conjunction with a provided visual example.

We recommend users to utilize this feature when they find a very **similar scene** to the sought one, or a scene that contains a **relevant concept**. To demonstrate this, without an explicit object counting module, it can be hard

Fig. 13: An image that can be used as a visual example for the example query. Interestingly, the photo was taken at the same spot with the correct one, albeit on another day.

for a model to differentiate between one, two, or more people in an image. We can use a visual example to encode the concept *having lunch with one person*, which is different from having two or zero companion. Furthermore, the visual example may carry additional information, such as the restaurant-implying surroundings, as opposed to home dining. This can be observed in Figure 13.

It is also important to convince users of the capabilities of this AI-based function, as it is different from conventional exact matching-based engines. For the general user, especially ones without an AI background, the underlying model is perceived as a black box, potentially preventing them from trusting in it. A good first step would be to show a lot of examples, together with some guiding principles.

6.4 Timeline viewing

Another important feature provided in FIRST is the Timeline View, which allows the user to view images in chronological order from a specific point in time defined by an image. As discussed in Section 3.5, this tool is indispensable as lifelogging data is naturally in time order, and humans by nature usually recall a sequence of events, as opposed to a single one.

We advise users to use this feature when there are **more than one events** mentioned and the user has found at least one in the chain. This feature is especially helpful for verification of the result, as while there can be many examples of a single event, the sequence of events is usually unique. For example, the image in Figure 14 can be found when backtracking from the groundtruth image. It can confirm that the photo indeed depicted the interior of a Subway, not any other restaurant. Also note that this feature can be combined with the visual similarity search in Section 6.3 to perform query expansion in both time and visual dimensions.

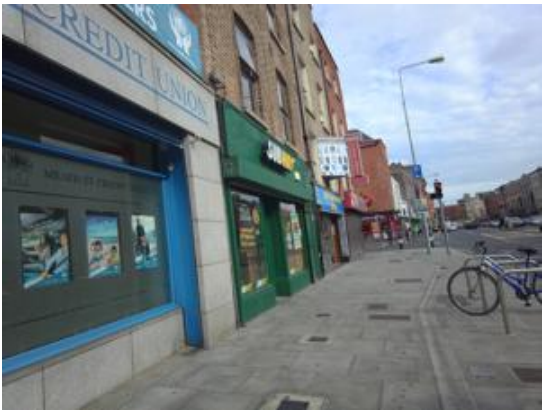

Fig. 14: An image of the Subway, found using the Timeline View feature.

7 Conclusion and Future Work

In this paper, we present our interactive retrieval system for lifelog exploration. This is the second version of our flexible retrieval platform FIRST[13]. Based on the user-interface and interactive platform, we can easily build various UI forms for users to interact with. With our system integration platform, we can add more query processing modules to our solution. Currently, our system consists of five main modules to handle query with meta-data, query with text description, scene clustering, query expansion, and temporal navigation.

To handle more flexible ways for user to interact with, our system needs to be gradually upgraded new functions and modalities that we can learn from other research teams and solutions. Furthermore, we intend to evaluate the efficiency of different interaction and query approaches to enhance the ease-of-use for our system.

Supplementary information. If your article has accompanying supplementary file/s please state so here.

Authors reporting data from electrophoretic gels and blots should supply the full unprocessed scans for key as part of their Supplementary information. This may be requested by the editorial team/s if it is missing.

Please refer to Journal-level guidance for any specific requirements.

Acknowledgments. This work was funded by Gia Lam Urban Development and Investment Company Limited, Vingroup and supported by Vingroup Innovation Foundation (VINIF) under project code VINIF.2019.DA19

Declarations

Some journals require declarations to be submitted in a standardised format. Please check the Instructions for Authors of the journal to which you are

submitting to see if you need to complete this section. If yes, your manuscript must contain the following sections under the heading 'Declarations':

- Funding
- Conflict of interest/Competing interests (check journal-specific guidelines for which heading to use)
- Ethics approval
- Consent to participate
- Consent for publication
- Availability of data and materials
- Code availability
- Authors' contributions

If any of the sections are not relevant to your manuscript, please include the heading and write 'Not applicable' for that section.

Editorial Policies for:

Springer journals and proceedings:
https://www.springer.com/gp/editorial-policies

Nature Portfolio journals:
https://www.nature.com/nature-research/editorial-policies

Scientific Reports:
https://www.nature.com/srep/journal-policies/editorial-policies

BMC journals:
https://www.biomedcentral.com/getpublished/editorial-policies

References

[1] Gurrin, C., Le, T.-K., Ninh, V.-T., Dang-Nguyen, D.-T., Jónsson, B.T., Lokoč, J., Hurst, W., Tran, M.-T., Schoeffmann, K.: An Introduction to the Third Annual Lifelog Search Challenge, LSC'20. In: ICMR '20, The 2020 International Conference on Multimedia Retrieval. ACM, Dublin, Ireland (2020)

[2] Gurrin, C., Schoeffmann, K., Joho, H., Zhou, L., Duane, A., Leibetseder, A., Riegler, M., Piras, L., Tran, M.-T., Lokoč, J., Hürst, W.: Comparing Approaches to Interactive Lifelog Search at the Lifelog Search Challenge (LSC2018). ITE Transactions on Media Technology and Applications **7**(2), 46–59 (2019)

[3] Heller, S., Gasser, R., Parian-Scherb, M., Popovic, S., Rossetto, L., Sauter, L., Spiess, F., Schuldt, H.: Interactive Multimodal Lifelog Retrieval with Vitrivr at LSC 2021. In: Proceedings of the 4th Annual on Lifelog Search Challenge. LSC '21, pp. 35–39. Association for Computing Machinery,

New York, NY, USA (2021). https://doi.org/10.1145/3463948.3469062. event-place: Taipei, Taiwan. https://doi.org/10.1145/3463948.3469062

[4] Trang-Trung, H.-P., Le, T.-C., Tran, M.-K., Ninh, V.-T., Le, T.-K., Gurrin, C., Tran, M.-T.: Flexible Interactive Retrieval SysTem 2.0 for Visual Lifelog Exploration at LSC 2021. In: Proceedings of the 4th Annual on Lifelog Search Challenge. LSC '21, pp. 81–87. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3463948.3469072. event-place: Taipei, Taiwan. https://doi.org/10.1145/3463948.3469072

[5] Nguyen, T.-N., Le, T.-K., Ninh, V.-T., Tran, M.-T., Thanh Binh, N., Healy, G., Caputo, A., Gurrin, C.: LifeSeeker 3.0: An Interactive Lifelog Search Engine for LSC'21. In: Proceedings of the 4th Annual on Lifelog Search Challenge. LSC '21, pp. 41–46. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3463948.3469065. event-place: Taipei, Taiwan. https://doi.org/10.1145/3463948.3469065

[6] Shin, J., Waldau, A., Duane, A., Jónsson, B.: PhotoCube at the Lifelog Search Challenge 2021. In: Proceedings of the 4th Annual on Lifelog Search Challenge. LSC '21, pp. 59–63. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3463948.3469073. event-place: Taipei, Taiwan. https://doi.org/10.1145/3463948.3469073

[7] Duane, A., Jónsson, B.: ViRMA: Virtual Reality Multimedia Analytics at LSC 2021. In: Proceedings of the 4th Annual on Lifelog Search Challenge. LSC '21, pp. 29–34. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3463948.3469067. event-place: Taipei, Taiwan. https://doi.org/10.1145/3463948.3469067

[8] Lokoč, J., Mejzlik, F., Veselý, P., Souček, T.: Enhanced SOMHunter for Known-Item Search in Lifelog Data. In: Proceedings of the 4th Annual on Lifelog Search Challenge. LSC '21, pp. 71–73. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3463948.3469074. event-place: Taipei, Taiwan. https://doi.org/10.1145/3463948.3469074

[9] Tran, L.-D., Nguyen, M.-D., Thanh Binh, N., Lee, H., Gurrin, C.: Myscéal 2.0: A Revised Experimental Interactive Lifelog Retrieval System for LSC'21. In: Proceedings of the 4th Annual on Lifelog Search Challenge. LSC '21, pp. 11–16. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3463948.3469064. event-place: Taipei, Taiwan. https://doi.org/10.1145/3463948.3469064

[10] Alam, N., Graham, Y., Gurrin, C.: Memento: A Prototype Lifelog Search

Engine for LSC'21. In: Proceedings of the 4th Annual on Lifelog Search Challenge. LSC '21, pp. 53–58. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3463948.3469069. event-place: Taipei, Taiwan. https://doi.org/10.1145/3463948.3469069

[11] Leibetseder, A., Schoeffmann, K.: LifeXplore at the Lifelog Search Challenge 2021. In: Proceedings of the 4th Annual on Lifelog Search Challenge. LSC '21, pp. 23–28. Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3463948.3469060. event-place: Taipei, Taiwan. https://doi.org/10.1145/3463948.3469060

[12] Tran, M.-T., Nguyen, T.-A., Tran, Q.-C., Tran, M.-K., Nguyen, K., Ninh, V.-T., Le, T.-K., Trang-Trung, H.-P., Le, H.-A., Nguyen, H.-D., Do, T.-L., Vo-Ho, V.-K., Gurrin, C.: FIRST - Flexible Interactive Retrieval SysTem for Visual Lifelog Exploration at LSC 2020. In: Proceedings of the Third Annual Workshop on Lifelog Search Challenge, pp. 67–72. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3379172.3391726

[13] Tran, M.-T., Nguyen, T.-A., Tran, Q.-C., Tran, M.-K., Nguyen, K., Ninh, V.-T., Le, T.-K., Trang-Trung, H.-P., Le, H.-A., Nguyen, H.-D., Do, T.-L., Vo-Ho, V.-K., Gurrin, C.: First - flexible interactive retrieval system for visual lifelog exploration at lsc 2020. In: Proceedings of the Third Annual Workshop on Lifelog Search Challenge. LSC '20, pp. 67–72. Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3379172.3391726. https://doi.org/10.1145/3379172.3391726

[14] Hoang-Xuan, N., Trang-Trung, H.-P., Nguyen, E.-R., Le, T.-C., Tran, M.-K., Ninh, V.-T., Le, T.-K., Gurrin, C., Tran, M.-T.: Flexible interactive retrieval system 3.0 for visual lifelog exploration at lsc 2022. In: Proceedings of the 2022 ACM Workshop on the Lifelog Search Challenge, LSC22, Newark, NJ (2022)

[15] Liu, Y., Chen, H., Shen, C., He, T., Jin, L., Wang, L.: Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)

[16] Le, N., Nguyen, D., Nguyen, V., Tran, M.: Lifelog moment retrieval with advanced semantic extraction and flexible moment visualization for exploration. In: Cappellato, L., Ferro, N., Losada, D.E., Müller, H. (eds.) Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum, Lugano, Switzerland, September 9-12, 2019. CEUR Workshop Proceedings, vol. 2380. CEUR-WS.org, ??? (2019). http://ceur-ws.org/Vol-2380/paper_139.pdf

[17] Le, N., Nguyen, D., Hoang, T., Nguyen, T., Truong, T., Duy, T.D.,

Luong, Q., Vo-Ho, V., Nguyen, V., Tran, M.: Smart lifelog retrieval system with habit-based concepts and moment visualization. In: Gurrin, C., Schöffmann, K., Joho, H., Dang-Nguyen, D., Riegler, M., Piras, L. (eds.) Proceedings of the ACM Workshop on Lifelog Search Challenge, LSC@ICMR 2019, Ottawa, ON, Canada, 10 June 2019, pp. 1–6. ACM, ??? (2019)

[18] Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. CoRR **abs/1506.01497** (2015) arXiv:1506.01497

[19] Tan, M., Pang, R., Le, Q.V.: Efficientdet: Scalable and efficient object detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10778–10787 (2020)

[20] Trang-Trung, H., Le, H., Tran, M.: Lifelog moment retrieval with self-attention based joint embedding model. In: Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (eds.) Working Notes of CLEF 2020 - Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, September 22-25, 2020. CEUR Workshop Proceedings, vol. 2696. CEUR-WS.org, ??? (2020). http://ceur-ws.org/Vol-2696/paper_60.pdf

[21] 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 answering. In: CVPR (2018)

[22] Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D.A., Bernstein, M., Fei-Fei, L.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. (2016). https://arxiv.org/abs/1602.07332

[23] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)

[24] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)

[25] Faghri, F., Fleet, D.J., Kiros, J.R., Fidler, S.: VSE++: improving visual-semantic embeddings with hard negatives. In: British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018, p. 12. BMVA Press, ??? (2018). http://bmvc2018.org/contents/papers/0344.pdf

[26] Lin, T., Maire, M., Belongie, S.J., Bourdev, L.D., Girshick, R.B., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. CoRR **abs/1405.0312** (2014) arXiv:1405.0312

[27] Le, T.-K., Ninh, V.-T., Tran, M.-T., Nguyen, T.-A., Nguyen, H.-D., Zhou, L., Healy, G., Gurrin, C.: Lifeseeker 2.0: Interactive lifelog search engine at lsc 2020. Association for Computing Machinery, New York, NY, USA (2020)

[28] Gurrin, C., Schoeffmann, K., Joho, H., Leibetseder, A., Zhou, L., Duane, A., Dang Nguyen, D.T., Riegler, M., Piras, L., Tran, M.-T., Lokoč, J., Hürst, W.: [Invited papers] Comparing Approaches to Interactive Lifelog Search at the Lifelog Search Challenge (LSC2018). ITE Transactions on Media Technology and Applications **7**, 46–59 (2019). https://doi.org/10.3169/mta.7.46