

A Multi-level Interactive Lifelog Search Engine with User Feedback

Jiayu Li
DCST, IAI, BNRist,
Tsinghua University
Beijing, China
lijayu16@mails.tsinghua.edu.cn

Min Zhang*
DCST, IAI, BNRist,
Tsinghua University
Beijing, China
z-m@tsinghua.edu.cn

Weizhi Ma
DCST, IAI, BNRist,
Tsinghua University
Beijing, China
mawz@tsinghua.edu.cn

Yiqun Liu
DCST, IAI, BNRist,
Tsinghua University
Beijing, China
yiqunliu@tsinghua.edu.cn

Shaoping Ma
DCST, IAI, BNRist,
Tsinghua University
Beijing, China
msp@tsinghua.edu.cn

ABSTRACT

With the rise of portable wearable devices, it is easier for users to save their lifelog data. As lifelog is usually disorganized with multi-modal information (even noisy sometimes), an interactive search engine is crucial for users to review and explore their lifelog. Unlike traditional search engines, lifelog search includes multi-modality information of images, text and other data from sensors, which brings challenges to data arrangement and search. Accordingly, users' information need is also multi-level. Hence, a single interaction mechanism may not be able to satisfy users' requirements. As the data set is highly personalized, interaction and feedback from users should also be considered in the search engine. Therefore, in this paper we present an interactive multi-modality lifelog search engine to help users manage and find lifelog data. To this end, lifelog data is clustered and processed in multi-level processing. Then, we build an interactive search engine, including *text as query*, *image as query*, and *timeline view* modules. Besides, the system is able to adopt user feedback mechanisms in multi-round queries. Our system shows promising experimental results on LSC'20 dataset and development topics. The text-based search module gives correct results on more than 60% of the development topics at LSC'20.

KEYWORDS

Lifelogging, Interactive Search Engine, Information Retrieval

ACM Reference Format:

Jiayu Li, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. A Multi-level Interactive Lifelog Search Engine with User Feedback. In *Proceedings of the Third Annual Workshop on the Lifelog Search Challenge (LSC '20)*,

*Contact author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

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

© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7136-0/20/06...\$15.00
<https://doi.org/10.1145/3379172.3391720>

June 9, 2020, Dublin, Ireland. ACM, New York, NY, USA, 7 pages. <https://doi.org/10.1145/3379172.3391720>

1 INTRODUCTION

Lifelogging is the process of passively gathering, processing and reflecting on life experience data collected by sensors [2]. With rapid development of portable devices, lifelogging becomes increasingly popular in people's daily life, and lifelog records reveal great potential in exploring and comprehending how we live our life.

As a way of recording daily life, lifelog includes tremendous but disorganized data. Sometimes people want to go through their records or search in the logs, which can help them recall details of certain scenes. Therefore, it is necessary to develop tools to organize lifelog data and enable users to search their records conveniently.

On the other hand, lifelog searching is a challenging problem. In lifelog searching, the photos are usually the search objects. However, traditional image search engines can not be directly used in the lifelog scenario, since the dataset organization and size are very different. Multi-modality information is recorded in lifelog through various sensors [9], which increase the difficulty of data processing and displaying. The multi-media information also requires multi-modally search modules to meet users' needs. Besides, lifelog is extremely personalized and users are usually more familiar with the dataset than in traditional search engine. Also, lifelog search is generally complex with noisy results, which needs users' clarification. Therefore, the search engine should be highly interactive.

To encourage researches in lifelog search applications, many related tasks and challenges have been proposed in recent years. *Lifelog Search Challenge (LSC)* is one of them, which aims at evaluating the performance of multi-media lifelog retrieval tools in live evaluation campaigns [4]. The challenge has held two successful iterations in 2018 and 2019. Compared with the past two iterations, this year the dataset expands from logs of less than one month to more than 100 days. The expanded dataset enlarges the searching space, as well as provides more complex spacial data, which brings more challenges for the accuracy of search engine. The abundant metadata information also provides more possibility for user queries, requiring more flexibility in user interaction.

To cope with the challenges in lifelog retrieval and perform the task in LSC'20, We propose a highly-interactive multi-modal lifelog

search engine. Compared with existing lifelog search engines, we generate multi-level features and consider user feedback in text-based queries. Besides, we also design image-based search and timeline view modules. Our main contributions are as follows:

- Multilevel features in lifelog records are generated and applied in our search engine, including image visual features, detail descriptions of the scene, and behavior expression.
- An interactive lifelog search engine is proposed, which is able to cope with not only textual queries but also image queries. A timeline view module is also included.
- Experiments with 24 topics are conducted to verify the utility of our search engine and promising results are shown.

2 RELATED WORK

As images are usually the main content and search target in lifelogging, lifelog search is related to traditional image retrieval researches. Image retrieval has been studied since the 1970s in computer vision and database management communities. At the early stage, text-based image retrieval was popular, in which images are first annotated with text, and then searched with text-based database [15]. With the development of digital images production and storage, using image queries to find similar images in database became necessary in large-scale dataset, called content-based image retrieval (CBIR) [5]. Meanwhile, user feedback is considered in interaction with image retrieval systems [14][11]. However, image retrieval works are mostly based on large-scale image datasets without context information but lifelog search is built on personal dataset, which is relatively small in size with rich metadata. Therefore, the lifelog search engines should pay more attention to user interaction and multi-modal data process.

With an increasing number of workshops and tasks on lifelogging, there are already a number of researches about lifelog search engines in recent years. In LSC'18 and LSC'19, many lifelog search engines have been proposed. Some of the systems are modifications of existing video search systems [7][1][8]. Others are designed specifically for lifelog search. In two-stage search engine [6], multi-modal image queries and faceted text queries were both supported in the system. The *Exquisitor system* proposed a novel way of simply using relevance feedback from user to find results [12]. In the *Lifelog Retrieval System V2* in LSC'19 [10], a tree structure is used for storing images, and answer reviewing module is employed for user feedback. All these systems show the nature of multi-modality and interaction in lifelog searching tasks. In our work, we also propose a multi-modal interactive search engine. What's more, we summarize a multi-level feature generation method to extract features from more metadata. And we introduce negative feedback for user interaction. We also add the timeline view module, which offers the user a well-structured overview of each day.

3 DATASET & FEATURE EXTRACTION

3.1 Dataset Overview and Preprocessing

The LSC'20 dataset is a new multi-modality dataset of 114 days' records collected in 2015, 2016, and 2018 [3]. It contains more than 190k images taken by a wearable camera, visual concepts detected in the images, and textual metadata about the collection. The metadata

includes detailed time, location, activity, and biometrics information about each picture. All data is collected from a lifelogger.

Since the photos are taken automatically, they can be blurry, blocked, and even repetitive. Therefore, we need to remove noises and duplicates from the images firstly. An OpenCV library in python¹ is applied to detect and remove 16,009 blurry and blocked images. Then, we choose the histogram similarity of the image's RGB representation as the similarity metric to detect duplication images:

$$s(H_a, H_b) = \frac{\sum_i (H_a[i] - \bar{H}_a)(H_b[i] - \bar{H}_b)}{\sqrt{\sum_i (H_a[i] - \bar{H}_a)^2 \sum_i (H_b[i] - \bar{H}_b)^2}} \quad (1)$$

where H_a and H_b are color histogram of two images, and \bar{H} is average of all histogram bins. After some pilot experiments and hand annotations, neighboring images with similarity over 0.85 are judged to be similar and clustered into a group. As photos in the same group are taken in similar scene, each group is called a *shot*. For each shot, the image with the most number of visual concepts is chosen as the *key image*. At last, we have 51,144 shots of denoised images, which is treated as the atom unit for feature generation and search, reducing the computation for our search engine greatly.

3.2 Multilevel Feature Extraction

The lifelog dataset is deeply multi-modal, continuous and potentially error-laden [2], so a clear structure is necessary to organize the data. We propose a three-level feature extraction method, in which visual, textual, and behavior features are extracted.

3.2.1 Visual Features. At the first level, visual features of shots are extracted. For each shot, the key image is divided into 3x3 parts, and **main color** of each part is classified into one of ten main colors: white, black, gray, red, orange, yellow, green, cyan, blue, and purple. **Lightness, contrast, and saturation** are also calculated from HSL (Hue, Saturation, Lightness) representation of the key image. Visual features are summarized in the first part of Table 1.

3.2.2 Textual Features. Visual concepts and metadata are extracted as textual features. Firstly, start time and end time for each shot are aligned in local and UTC time zone according to metadata information. Then, biometrics data for shot is extracted and calculated, of which **calories** and **steps** are represented by summation, while **elevation, speed** and **heart rate** are averaged in shot.

Another important part of textual features is **objects**. Besides visual concepts in the initial LSC'20 dataset, we use a state-of-art object detection network, faster r-cnn [13] (pre-trained on COCO dataset) to detect objects in each image. To get more descriptions about the objects, *DenseCap API* from deepai² is used to label the color or texture of each object. Finally, **objects** detected by three ways are merged and deduplicated, and objects with confidence score higher than 0.5 are retained. Considering the processing time, only the *key image* in each shot is detected by this API.

The second part of Table 1 presents textual features.

3.2.3 Behavior Features. At the highest level, users' behavior features are extracted. In the metadata file, we have time zone, location, and coordinate of user in each minute. However, some locations

¹<https://pypi.org/project/opencv-python/>

²<https://deepai.org/>

Table 1: Name, source, type and description of generated features in three levels.

Feature	Source	Type	Description
Visual Features			
Main Color	images	list	List of main color for 3*3 parts of key image
Lightness	images	float	Brightness of key image
Contrast	images	float	Contrast of key image
Saturation	images	float	Saturation of key image
Textual Features			
Local start time	metadata	datetime	Start time of shot in local time zone
Local end time	metadata	datetime	End time of shot in local time zone
UTC start time	metadata	datetime	Start time of shot in UTC time
UTC end time	metadata	datetime	End time of shot in UTC time
Elevation	metadata	float	Average elevation
Calorie	metadata	float	Total calorie intake
Step	metadata	int	Total steps
Speed	metadata	float	Average speed
Heart rate	metadata	float	Average heart rate
Objects	visual concepts, object detection model and API	list	Objects seen in the scene
Behavior Features			
Locations	metadata and manual expansion	string	Possible location during the shot
Activities	metadata	string	Activity labeled in data set

are missing. So we collect the longitude and latitude for the known locations, and fill in the missing locations by their coordinates. If the location is still undefined and the lifelogger was not moving (judged from biometrics), the location will be tagged same as past. The few undefined locations are labeled as ‘unknown’. To better use the locations, we expand and clarify **location** manually with the help of google map, and time zone is also taken as the overall location. Meanwhile, **activity** is labeled for each shot as well.

Behavior features are summarized in the third part of Table 1.

3.3 Contextual information

In practice, it is natural for users to recall events before or after the scene when searching for a certain time point. So we add context information based on features generated for each shot. To be specific, **locations**, **activities**, and **objects** in past or future an hour are collected as context features for a shot, named as **past** and **future** features respectively.

4 INTERACTIVE SEARCH ENGINE

In this section, we introduce search mechanism and interactive interface of our system. In the lifelog search process, users may generally recall the target scene and repeatedly modify the query. So we build a multi-round highly-interactive *text as query* search

module. Besides text, it is also necessary to provide a way for users to search for similar images. For example, when a user wants to know the last time he/she went to a certain restaurant, he/she can take a picture of the restaurant, and search for similar scenes in his/her records. Therefore, we further design a search component with *images as query*. In addition, to give users a clearer way to understand lifelog in general, a *timeline view* module is also provided for user. The framework of our system is shown in Figure ??.

4.1 Search Mechanism

In our system, each shot, clustered with the methods in Section 3.1, is represented as a document. Each document contains abundant visual, textual and behavior features described in Table 1, which are extracted from images and metadata information with the multi-level feature extraction approach described in Section 3.2. Each feature corresponds to a facet in search engine schema. Whoosh³ is used for building our search modules, and inverted index is employed to save documents. We also build a dictionary of **time**, **objects**, **location**, and **activity** features in our dataset. To cope with timeline view module, for each day in lifelog, neighboring shots with the same **location** are clustered as a time period, and objects with the highest confidence score are saved for each timeline period. The offline process is shown in the middle of Figure ??.

In online search process of *text as query* module, given a textual query, it is first parsed into facets and looked up in the feature dictionary to form a faceted query automatically. All the facets are combined with logical AND, while logical OR is used to connect multiple query keywords within locations, activities, and objects due to noises in these features. With the faceted query, BM25F scoring function is then performed to search for top N relevant shots, and results are listed in descending order of their relevance scores. If the user further gives a modified query with negative feedback at result list, the search engine will extract images and keywords in feedback and exclude them in the following search with logical NOT. Then, the search engine will present a new result list. The flow path may continue until user finds the correct answer, which is shown in the bottom left of Figure ??.

For image query, as user inputs a query image, the system will extract visual features and detail descriptions using methods mentioned in Section 3.2.1 and Section 3.2.2. In timeline view module, all information of the searched date interval will be displayed on the website page as daily view of the user. The searching method for timeline view, and searching and ranking methods for image query are the same as the textual query module.

4.2 Search Engine Interface

In the following subsection, we demonstrate the interface and interactive process of three modules in our system, which makes the search engine convenient to use for both expert and novice users.

4.2.1 Text as query. As shown in Figure 2(a), in *text as query* search module, user can type query in textbox and click the ‘Fill in forms’ button. The query parser of system will then parse and fill in the

³<https://pypi.org/project/Whoosh/>

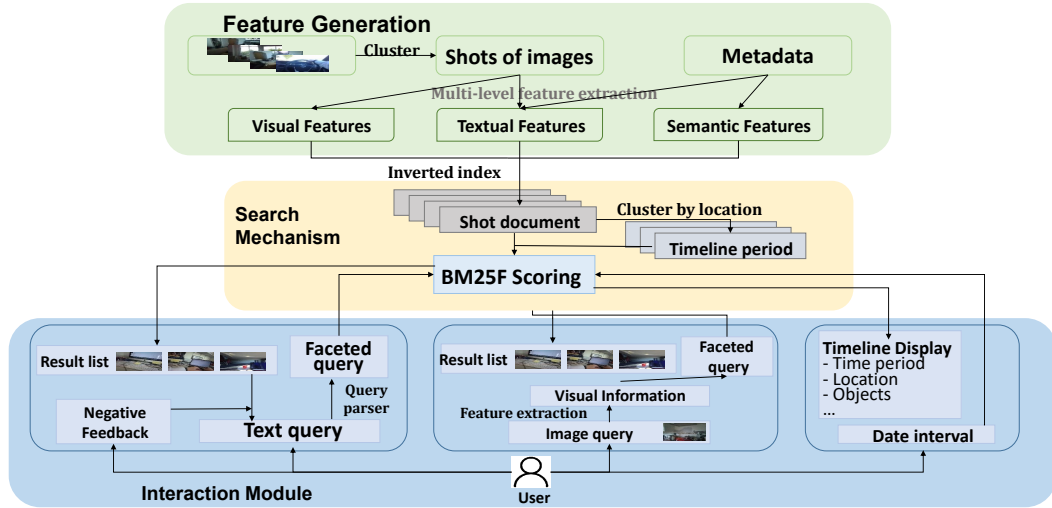


Figure 1: Framework of our lifelog search engine.

following faceted forms on the page automatically using the mechanism in Section 4.1. User can add, delete or modify any facets of the query in an interactive way.

Once user submits the query, a result page containing top N related shots (Figure 2(b)) will be returned. On the left, the query text and faceted query are presented for further modification. The right part is result list containing the key images of shots, metadata information and user feedback module. When user hovers the mouse on the image, biometrics and objects in the shot will appear.

If user wants to continue with this query, he/she can modify the query on the left part. For each shot in the ranking list, user can give negative feedback by choosing to exclude this shot or similar shots. User can also exclude images with certain objects by filling in the text box. The selected objects, images, time, or location will not display in the following result lists. By the feedback, we can acquire more information need from the user that may not be fully described by designed query facets.

4.2.2 Image as query. In this module, user can upload an image as query, and choose to find photos similar in content or semantics (or both). The interface is shown in Figure 3. Upon clicking on ‘search’, user can get a result list of images shown in the same way as *text as query* result page (Figure 2(a)), but without the feedback mechanism.

4.2.3 Timeline view. To give user a clearer way to understand lifelog in general, we add a *timeline view* module (displayed in Figure 4). At the top of page, user can input a date interval. Then the location tracking, dwell time and relevant objects of each day in the interval will be displayed. User can further click on any location to view key images of the shots at that location in the corresponding time period.

5 EXPERIMENTS AND RESULTS

In this section, we verify the *text as query* module of our search engine on the development topics at LSC’20. In every topic, 6 increasingly detailed descriptions are given step by step, revealed every 30 seconds. Therefore, we construct queries in a sequential

way, submitting a query every 30 seconds. In our experiment, each result list contains 10 shots.

5.1 Example Query

Taken the first topic in development topics, topic 25, as an example, we perform a multi-round interactive search. The search process is recorded in Figure 5. We can observe from the second and third lines that the automatic query parser is able to extract keywords from the textual query correctly. The usefulness of negative feedback also appears over time. For example, from 30s to 60s, with the same faceted query, the rank of the accurate answer is lifted from third to first due to feedback.

5.2 Overall Result

We then test our search engine with 24 development topics at LSC’20. On the user side, the interfaces described in Section 4.2.1 and Section 4.2.2 are used. We input the textual query, let the system parse query into facets, and search without any changes in faceted query, except adding a few locations that the system fails to detect automatically in topic 27, 32, 34, 42, and 43. On the system side, our search engine is built on the four-month dataset released in LSC’20 and utilize the mechanisms described in Section 4.1.

Hit ratio is adopted as the evaluation metric for performance. Result lists that contain at least a relevant result is considered a hit.

The overall performance of our system is shown in Table 2. It is observed that as query description becomes more detailed, the faceted query contains more keywords, and the hit ratio is increasing. At 150s, more than 60% of queries are correctly found in top 10 results, showing promising performance of our search engine. Besides, we can see a remarkable promotion in performance in last three stages of time. It may be associated with that most of the key information is given after 90s. Explicit **location** or **weekday** are usually given at 120s or 150s, which filter irrelevant results efficiently. Another strong filter that often appears after 90s is **time of day**. Accurate detection of objects in the scene also helps a lot. For example, ‘Unix book’ in topic 41 is exactly detected in relevant

Text Input

Find the time when I was looking at an old clock, with flower visible.

Fill In Forms

Basic Information

Color of the image:

brightness: low: medium: high:

contrast: low: medium: high:

saturation: low: medium: high:

Textual Description

Time	Objects	Biometrics Logs
Local Time Date: <input type="text"/> Weekday: <input type="text"/> Approximate time: <input type="text"/> Range of hour: <input type="text"/>	Objects in the scene: old clock flower Add Objects seen in future 1 hour: Add Objects seen in past 1 hour: Add	Total Elevation is above <input type="text"/> Total calorie is above <input type="text"/> Total Step is above <input type="text"/> Average speed is above <input type="text"/> Average heart rate is above <input type="text"/> <small>*if 'between' is chosen, please input two numbers and separate by ..</small>

Semantic Description

Description in this shot	Description for future 1 hour	Description for past 1 hour
Location: <input type="text"/> Activity: <input type="text"/>	Location: <input type="text"/> Activity: <input type="text"/>	Location: <input type="text"/> Activity: <input type="text"/>

Search

(a) Search Page

Text Input

Find the time when I was looking at an old clock, with flower visible.

Fill In Forms

Basic Information

Color of the image:

Image basic parameters: brightness-> low: medium: high:
contrast-> low: medium: high:
saturation-> low: medium: high:

Detail Description

Local Time
Date:
Weekday:
Approximate time:
Range of hour:

Objects
Objects in the scene:
old clock
flower
Add
Objects seen in future 1 hour:
Add
Objects seen in past 1 hour:
Add

Semantic Description

Description in this shot
Location:
Activity:

Negative Feedback

Image Feedback
exclude 20160825_223717_000.jpg
exclude 20160824_341827_000.jpg
exclude 20160925_190037_000.jpg

Exclude Images with tag:

Search again

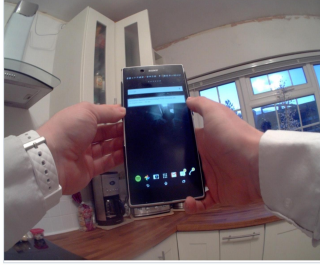
(b) Result Page

Figure 2: Interface for Text as query module.

Image as Query

Choose File: No file chosen

Upload the image



Find images similar in
content (color and hue)
semantics (objects)

Search

Figure 3: Interface for Image as query module.

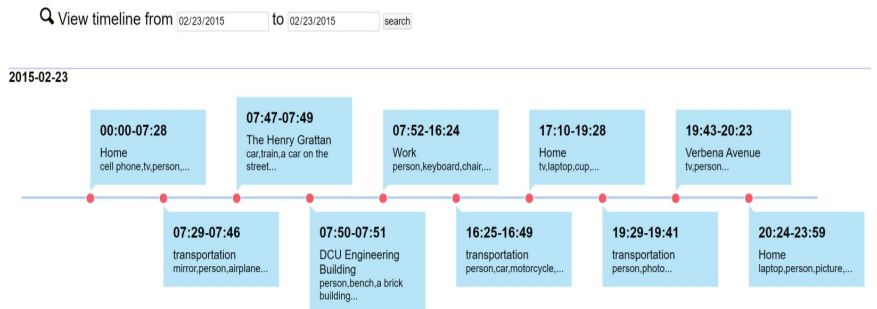


Figure 4: Interface for Timeline view module.

Table 2: Performance of text as query module on 24 development topics.

	hit@1	hit@3	hit@5	hit@10
0s	0.04	0.04	0.04	0.04
30s	0.04	0.04	0.04	0.04
60s	0.04	0.04	0.04	0.08
90s	0.04	0.08	0.13	0.33
120s	0.04	0.25	0.33	0.50
150s	0.17	0.29	0.46	0.63

images, which lead to rank first of correct answer at 0s. On the other hand, the hit ratio before 60s is low and slow-growing because the description at first is general. For example, queries like 'I was in the bank' match many scenes in the four-month logs, which may result in low ranking of correct answers at the beginning.

Although we achieve 0.63 on hit@10 at 150s, which means our system performs well on 15 topics, there are still 9 topics missed in the experiment. Inspecting these topics, we summarize reasons for mismatching as follows. Firstly, key objects from the queries may not be detected in the relevant images. For instance, 'a white dolls house' in topic 28 and 'coffee machine' in topic 36 are not detected in the relevant results, leading to search failure of these topics. Secondly, behavior features (location and activity) are insufficient for some queries. Taken topic 37 as an example, 'I remember I was washing clothes', activity 'washing' is left out as we have no relevant labels in dataset. This expands the search range greatly and makes it hard to find answers. Besides, as the topics are designed for the one-month dataset in LSC'19, some queries may be ambiguous in the enlarged dataset this year.

For the computing time, the text query parser takes about 1 second to parse the query, and retrieval time is around 3 seconds.

Time	0s	30s	60s	90s	120s	150s
Query	Find the time when I was looking at an old clock, with flowers visible.	Find the time when I was looking at an old clock, with flowers visible. There was a lamp also.	Find the time when I was looking at an old clock, with flowers visible. There was a lamp also, and a small blue monster (perhaps a long rabbit) watching me.	Find the time when I was looking at an old clock, with flowers visible. There was a lamp also, and a small blue monster (perhaps a long rabbit) watching me. Maybe there were two monsters.	Find the time when I was looking at an old clock, with flowers visible. There was a lamp also, and a small blue monster (perhaps a long rabbit) watching me. Maybe there were two monsters. It was a Monday or a Thursday.	Find the time when I was looking at an old clock, with flowers visible. There was a lamp also, and a small blue monster (perhaps a long rabbit) watching me. Maybe there were two monsters. It was a Monday or a Thursday. I was at home and in a bedroom.
Parsed Query (Automatically)	objects: flowers, old clock	objects: flowers, old clock, lamp	objects: flowers, old clock, lamp, small blue monster	objects: flowers, old clock, lamp, small blue monster	objects: flowers, old clock, lamp, small blue monster weekday: Monday, Thursday	objects: flowers, old clock, lamp, small blue monster, bedroom weekday: Monday, Thursday location: bedroom, home
Result	20160928_071412_000.jpg 20160921_181336_000.jpg 20160919_053811_000.jpg 20160813_155703_000.jpg 20160903_231317_000.jpg 20160904_115144_000.jpg 20160826_184908_000.jpg 20160901_055616_000.jpg 20160914_230510_000.jpg 20180520_074046E.jpg	20160921_181336_000.jpg 20160831_205608_000.jpg 20160919_053811_000.jpg 20160822_061808_000.jpg 20160925_190037_000.jpg 20160925_190037_000.jpg 20180527_181500E.jpg 20160903_231317_000.jpg 20160924_184249_000.jpg 20160904_115144_000.jpg 20180512_140910E.jpg	20160831_205608_000.jpg 20160919_053811_000.jpg 20160822_061808_000.jpg 20160925_190037_000.jpg 20160903_231317_000.jpg 20160904_115144_000.jpg 20180512_140910E.jpg 20160911_210949_000.jpg 20180503_185530E.jpg 20180520_203554E.jpg	20160822_061808_000.jpg 20160925_190037_000.jpg 20160903_231317_000.jpg 20160911_210949_000.jpg 20180526_163014E.jpg 20180506_085314E.jpg 20180529_183539E.jpg 20180515_123850E.jpg 20160822_202254_000.jpg 20160920_220620_000.jpg	20160822_061808_000.jpg 20160925_190037_000.jpg 20160903_231317_000.jpg 20160911_210949_000.jpg 20180530_202341E.jpg 20180514_234106E.jpg 20160911_093631_000.jpg 20180515_202733E.jpg 20160912_153257_000.jpg 20160914_230348_000.jpg	20160822_061808_000.jpg 20160811_064303_000.jpg 20160818_151402_000.jpg 20160926_064505_000.jpg 20160822_193037_000.jpg 20160915_194940_000.jpg 20180507_120343E.jpg 20180507_122832E.jpg 20160822_061912_000.jpg 20180503_173841E.jpg
Negative Feedback	exclude 20160813_155703_000.jpg, 20160826_184908_000.jpg, 20160901_055616_000.jpg, 20160914_230510_000.jpg, 20180520_074046E.jpg	exclude 20160921_181336_000.jpg, 20160924_184249_000.jpg, 20180527_181500E.jpg	exclude 20160831_205608_000.jpg, 20160919_053811_000.jpg, 20160904_115144_000.jpg, 20180512_140910E.jpg, 20180503_185530E.jpg	exclude 20160822_202254_000.jpg, 20160920_220620_000.jpg, 20180526_163014E.jpg, 20180506_085314E.jpg, 20180529_183539E.jpg	exclude 20160914_230348_000.jpg, 20180514_234106E.jpg, 20180530_202341E.jpg exclude images similar with 20160912_153257_000.jpg in location	

Figure 5: Interactive search process of topic25 in our system.(Correct answer is marked bold in the fourth row.)

Table 3: Performance of text as query module at 24 development topics with user feedback.

	hit@1	hit@3	hit@5	hit@10
0s	0.04	0.04	0.04	0.04
30s	0.04	0.04	0.08	0.08
60s	0.04	0.13	0.21	0.25
90s	0.17	0.21	0.29	0.50
120s	0.25	0.29	0.42	0.58
150s	0.42	0.46	0.58	0.63

(All the computing times are calculated for an interface running on a server with Intel i7-5930K CPU)

5.3 Impact of Feedback

We further test the impact of user feedback mechanism in our system. As the user feedback is highly interactive, we simulate an interaction situation where an expert user gives negative feedback to the first result (if it is wrong) and results that are irrelevant to query obviously, using the interface displayed in Figure 2(b).

The results are displayed in Table 3. Comparing Table 2 and Table 3, we can see a noticeable improvement in performance. With user feedback, the relevant results are detected earlier in the experiment. Hit@10 achieves 0.25 at 60s, and hit ratio at 90s are comparable to results at 120s without feedback. It shows that some relevant results are low ranking at the beginning, because too many images match the parsed query when it is not explicit, but we filter some irrelevant images based on our comprehension of the query. This indicates the information gap between original query and parsed query, but also demonstrates the necessity and usefulness of user feedback in lifelog search scenario. Meanwhile, the target results are ranking higher with user feedback. For instance, hit@1 at 150s promotes significantly. It is a natural result of removing irrelevant

images with high rank in negative feedback process. However, the mismatched 9 queries are still not found in the experiment with user feedback. User feedback mechanism is not able to completely solve the shortcomings in feature extraction and pre-indexing.

6 CONCLUSION AND FUTURE WORK

In this paper we present our multilevel interactive lifelog search engine with user feedback, which contains three components, *text as query*, *image as query*, and *timeline view*.

We first propose a multi-level feature generation method to extract information from the multi-modal dataset. Then a highly-interactive search engine is designed, where user can find images with textual/image queries and view daily timelines in lifelog. The search engine is implemented with a user-friendly interface, so that our system can be used by both expert and novice users.

The experiments on development topics show promising results of our system. More than 60% of the topics are hit at the end of query, which verifies the effectiveness of our automatic query parser and search methods. And further experiments indicate the usefulness of user feedback mechanism.

For future work, optimizing methods to eliminate noise in dataset can be a direction. More tools from computer vision and natural language processing communities will be used for object detection and semantic information extraction. Moreover, a better query parser in *text as query* will also be considered to fill the gap between textual and faceted query to generate more precise results.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (2018YFC0831900) and Natural Science Foundation of China (Grant No. 61672311, 61532011). Dr Weizhi Ma has been supported by Shuimu Tsinghua Scholar Program.

REFERENCES

- [1] Münzer Bernd, Leibetseder Andreas, Kletz Sabrina, Primus Manfred Jürgen, and Schoeffmann Klaus. 2018. lifeXplore at the lifelog search challenge 2018. In *Proceedings of the 2018 ACM Workshop on The Lifelog Search Challenge*. 3–8.
- [2] Gurrin Cathal, Smeaton Alan F, Doherty Aiden R, et al. 2014. Lifelogging: Personal big data. *Foundations and Trends® in information retrieval* 8, 1 (2014), 1–125.
- [3] Gurrin Cathal, Le Tu-Khiem, Ninh Van-Tu, Dang-Nguyen Duc-Tien, Jónsson Björn Þór, Lokoč Jakub, Hurst Wolfgang, Tran Minh-Triet, and Schoeffmann Klaus. 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.
- [4] Dang-Nguyen Duc-Tien, Schoeffmann Klaus, and Hurst Wolfgang. 2018. LSE2018 Panel-Challenges of Lifelog Search and Access. In *Proceedings of the 2018 ACM Workshop on The Lifelog Search Challenge*. 1–2.
- [5] Long Fuhui, Zhang Hongjiang, and Feng David Dagan. 2003. Fundamentals of content-based image retrieval. In *Multimedia information retrieval and management*. Springer, 1–26.
- [6] Nguyen Van Khan Isadora, Shrestha Pranita, Zhang Min, Liu Yiqun, and Ma Shaoping. 2019. A Two-Level Lifelog Search Engine at the LSC 2019. In *Proceedings of the ACM Workshop on Lifelog Search Challenge*. 19–23.
- [7] Lokoč Jakub, Souček Tomáš, Čech Premysl, and Kovalčík Gregor. 2019. Enhanced VIRET tool for lifelog data. In *Proceedings of the ACM Workshop on Lifelog Search Challenge*. 25–26.
- [8] Rossetto Luca, Gasser Ralph, Heller Silvan, Amiri Parian Mahnaz, and Schuldt Heiko. 2019. Retrieval of structured and unstructured data with vitivr. In *Proceedings of the ACM Workshop on Lifelog Search Challenge*. 27–31.
- [9] Dodge Martin and Kitchin Rob. 2007. 'Outlines of a world coming into existence': pervasive computing and the ethics of forgetting. *Environment and planning B: planning and design* 34, 3 (2007), 431–445.
- [10] Le Nguyen-Khang, Nguyen Dieu-Hien, Hoang Trung-Hieu, Nguyen Thanh-An, Truong Thanh-Dat, Dinh Duy-Tung, Luong Quoc-An, Vo-Ho Viet-Khoa, Nguyen Vinh-Tiep, and Tran Minh-Triet. 2019. Smart lifelog retrieval system with habit-based concepts and moment visualization. In *Proceedings of the ACM Workshop on Lifelog Search Challenge*. 1–6.
- [11] Vasconcelos Nuno and Lippman Andrew. 2000. Learning from user feedback in image retrieval systems. In *Advances in neural information processing systems*. 977–986.
- [12] Khan Omar Shahbaz, Jónsson Björn Þór, Zahálka Jan, Rudinac Stevan, and Worring Marcel. 2019. Exquisitor at the lifelog search challenge 2019. In *Proceedings of the ACM Workshop on Lifelog Search Challenge*. 7–11.
- [13] Ren Shaoqing, He Kaiming, Girshick Ross, and Sun Jian. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.
- [14] Guo Xiaoxiao, Wu Hui, Cheng Yu, Rennie Steven, Tesauo Gerald, and Feris Rogerio. 2018. Dialog-based interactive image retrieval. In *Advances in Neural Information Processing Systems*. 678–688.
- [15] Rui Yong, Huang Thomas S, and Chang Shih-Fu. 1999. Image retrieval: Current techniques, promising directions, and open issues. *Journal of visual communication and image representation* 10, 1 (1999), 39–62.