LifeMon: A MongoDB-Based Lifelog Retrieval Prototype

Alexander Christian Faisst
IT University of Copenhagen
Copenhagen, Denmark
achf@itu.dk

Björn Þór Jónsson
IT University of Copenhagen
Copenhagen, Denmark
bjth@itu.dk

ABSTRACT

We present LifeMon, a new lifelog retrieval prototype targeting LSC. LifeMon is based around the MongoDB document store, which is one of a host of scalable NoSQL systems developed over the last two decades, with a semi-structured data model that seems well matched with lifelog requirements. Preliminary results indicate that the system is efficient and that novice users can successfully use it to solve some LSC tasks.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; *Search engine architectures and scalability*; *Information retrieval query processing*; **Information storage systems**.

KEYWORDS

Lifelogging, Document Store, MongoDB, LifeMon, Scalability

ACM Reference Format:
Alexander Christian Faisst and Björn Þór Jónsson. 2021. LifeMon: A MongoDB-Based Lifelog Retrieval Prototype. In *Proceedings of the 4th Annual Lifelog Search Challenge (LSC '21), August 21, 2021, Taipei, Taiwan.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3463948.3469066

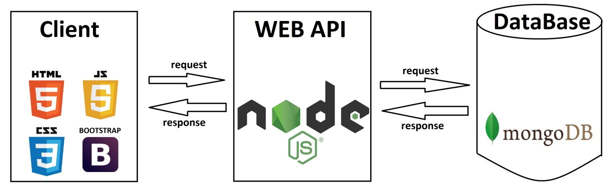


Figure 1: LifeMon's architecture consists of a web-based user interface, a Web-API and a MongoDB document store.

1 INTRODUCTION

Lifelogging is a branch of multimedia analytics that has recently seen growing interest in the research community [6, 7]. A lifelog is a collection of data centered on a single person, typically consisting of (a) a stream of images, often 1-3 per minute, along with image metadata, and (b) various sensor readings, such as heartbeat and motion data, that can be indirectly associated with the images. As such, a lifelog can be considered a semi-structured collection of moments in a person's life, which can be used for a variety of tasks; often categorised into five "R"s: recollecting; reflecting, reminiscing, remembering intentions, and retrieving [13].

The Lifelog Search Challenge (LSC) is a competition focusing on retrieval from a standard lifelog collection [6]. Since its inception in 2018, the competition has grown to include more than a dozen systems with a variety of approaches to lifelog retrieval [5, 8]. Many of the LSC systems focus on advanced multimedia techniques for analysing the collection, e.g., analysing the collection to detect events [14]. Other systems explore new modes of interaction beyond traditional search, e.g., interactive learning [11, 12] or virtual reality [2, 3].

Because the LSC collection is very small, only consisting of about 190K images corresponding to 4 months in the lifelogger's life, the scalability of data access is not a major consideration of the competition. While some systems claim scalability to much larger collections (e.g., [9, 10]), the LSC collection is simply too small to exercise scalability. Quick back-of-the-envelope calculations, however, show that even a single lifelog can become a significantly large collection over a lifetime, indicating that maintaining a personal lifelog system would be beyond the capabilities of most lifeloggers. And, correspondingly, providing successful lifelog services to the general public could result in massive collections of lifelog data. Thus, scalability of data access is an issue worth considering [4].

Over the last couple of decades, the database community has developed many scalable approaches. In particular, several classes of distributed and scalable non-relational systems have been developed under the umbrella term of NoSQL systems. The class of NoSQL systems that is seemingly most relevant to lifelog data is that of document stores; systems that organise their collections as a set of documents—typically defined using JSON or XML—with capabilities for accessing the document collection using a declarative query language and indexing the collection for performance.

In this paper, we present LifeMon, a new lifelog retrieval prototype based on MongoDB, the most popular open-source document store [1]. Figure 1 shows the overall architecture of LifeMon. We treat each lifelog image, along with all its relevant metadata and censor data, as a pair of documents. The user interface then provides (a) various filters to reduce the collection to a small subset, and (b) a timeline explorer to study the resulting images in context. A preliminary user study, based on a sub-set of the LSC 2019 tasks, with participants ranging from novices to LSC experts, indicates that the system is a viable LSC participant. Furthermore, performance measurements show that MongoDB can answer queries over the LSC collection quite efficiently on low-end hardware.

LSC '21, August 21, 2021, Taipei, Taiwan

ACM ISBN 978-1-4503-8533-6/21/08…$15.00
https://doi.org/10.1145/3463948.3469066

```
lsc2020-meta_data.csv (Administrative Data)
Minute_id:       20150223_0706
UTC time:        UTC_2015-02-23_07:06
Local time:      2015-02-23_07:06
Timezone:        Europe/Dublin
Lat:             53.38909481
Lon:             6.15800409
Semantic name:   Home
Elevation:       77.0
Speed:           0.0
Heart:           NULL
Calories:        1.2062000036239624
Activity type:   NULL
Steps:           NULL

lsc2020-visual_concepts.csv (Descriptive Data)
Minute_id:       20150223_0706
Local time:      2015-02-23_07:06
Image path:      DATASETS/LSC2020/2015-02-23/b00000000_21i6bq_20150223_070647e.jpg
Attributes:      nohorizon, enclosed area, man-made, dry, indoor lighting,
                 natural light, cloth, wood, glass, matte
Categories:      dorm_room - 0.144,  jail_cell - 0.111,  alcove - 0.092,
                 berth - 0.074,  hospital_room - 0.07
Concepts:        bed - 9833653569221495
```

Figure 2: An example LSC image, along with its metadata.

```
minute_id: {
    type: String,
    index: true
},
utc_time: String,
local_time: String,
time : {
    year: int,
    month: int,
    day: int,
    time_of_day: String,
    index: true
},
location : {
    semantic_name: String,
    timezone: String,
    lat: Number,
    lon: Number,
    elevation: Number
},
bio: {
    speed: Number,
    heart: Number,
    calories: Number,
    activity_type: String,
    steps: Number
}
```

(a) Administrative / sensor data.

```
minute_id: {
    type: String,
    index: true
},
local_time: String,
date: Date,
image_path: String,
colors: {
    dominant_color:
        [int,int,int],
    palette: [[int,int,int]]
},
attributes: {
    type: [ String ],
    index: true
},
categories: {
    type: [ {
        "_id": false,
        "name": String,
        "score": Number
    } ],
    index: true
},
concepts: {
    type: [ {
        "_id": false,
        "class": String,
        "score": String,
        "bbox": String
    } ],
    index:true
}
```

(b) Semantic / visual metadata.

Figure 3: Document schemas for the LifeMon database.

2 THE MONGODB DATABASE

The data provided by the LSC consists of images and metadata associated with these. The images are depictions of tasks performed throughout the day, such as watching television, grocery shopping, garden work, or sitting in a café. There are more than 190K images, taken over a total of four months, in the years 2016 to 2018. In this section, we first outline the LSC collection, and then describe how it is represented in the LifeMon document database.

2.1 LSC Metadata

Figure 2 shows an example image, along with its metadata entries. As the figure shows, the metadata is split into two files, which link the images together by a minute_id attribute. One file contains primarily administrative data, such as geo-coordinates and timestamps, as well as data like heart rate and calorie usage at the time when the picture was taken. The second file contains descriptive data, relating to the actual contents of the image, which is divided into three general types: *Attributes* describing the environment depicted in the image, e.g., lighting, indoor/outdoor, and natural/man-made; *Categories* describing the location the picture was taken in, e.g., kitchen, bedroom, office, or garden; and *Concepts* describing the visual contents of the image, e.g., cup, person, car, dog. The latter two categories are accompanied by a confidence score.

2.2 Document Store Schema

We mapped the contents of the two files to two different document collections. Each image thus corresponds to two documents, which are joined on the minute_id attribute. We also extracted color information from the images and added to the semantic metadata, and grouped related concepts into sub-documents. Figures 3a and 3b show the document schema definitions of these two files, respectively. For better performance, we indexed all the semantic attributes; this is illustrated by the ‘index: true” lines in Figure 3.

2.3 Ranking of Results

The administrative data is queried first, and then the narrowed-down result is used as a base for a second query to the semantic data. If multiple values are given in the query for a particular document attribute, these are OR-ed together, while filters on different document attributes are AND-ed together. To be returned, the documents therefore must match at least one value for each of the specified attributes. The documents that pass through the filters are ordered first by the total number of attribute values that match the query, and second by their document ID.

2.4 Performance Evaluation

We evaluated LifeMon’s performance times when solving LSC tasks, using the same tasks as used in the user evaluation reported below.

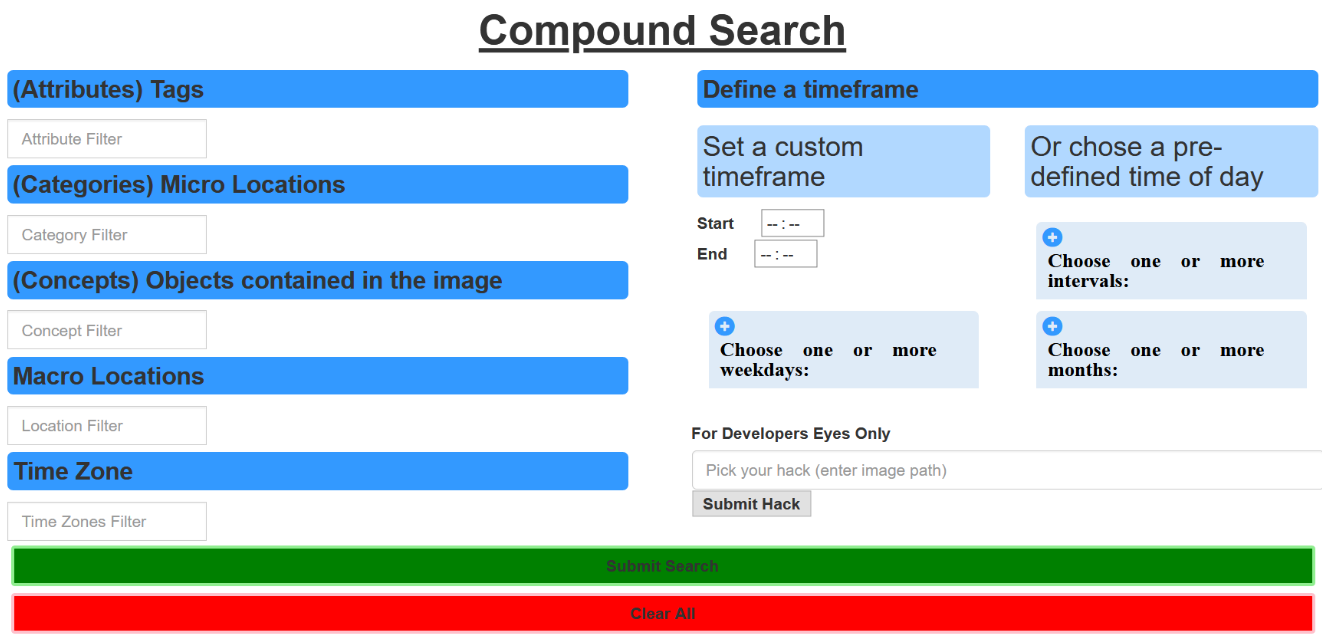


Figure 4: The main user interface of LifeMon, with keyword filters for various attributes on the left and a variety of time-related filters on the right.

All experiments were run on an otherwise idle desktop PC with a 3.2 GHz 6-Core Intel Core i7 processor with 32 GB 3200 MHz DDR4 RAM, running MS Windows 10 Home 10.0.19042 (19042) and MongoDB version 4.0.9. Both the Web-API and the database were running locally, and indexes were defined.

The results are shown in Table 1. The table first shows the total time from the time the request is sent from the client until it receives a response from the Web-API. It also shows separate query times for the individual collections of Figure 3. Lastly, the table also shows how many total entries/filter-inputs were part of the request, and over how many separate overall database fields these were spread (e.g. concepts, weekdays, months, etc), as well as the number of results returned. The number of entries/fields used are based on a search result with a meaningful/realistic precision.

Table 1 shows that the total query time ranges from 0.05 to 1 seconds. The table also shows that the total MongoDB query time never exceeds 0.7 seconds. The remainder of the time, which is spent on communications, is somewhat proportional to the result size. Overall, this performance is adequate for solving LSC tasks.

Table 1: Query performance for LifeMon and MongoDB. All measurements are in seconds.

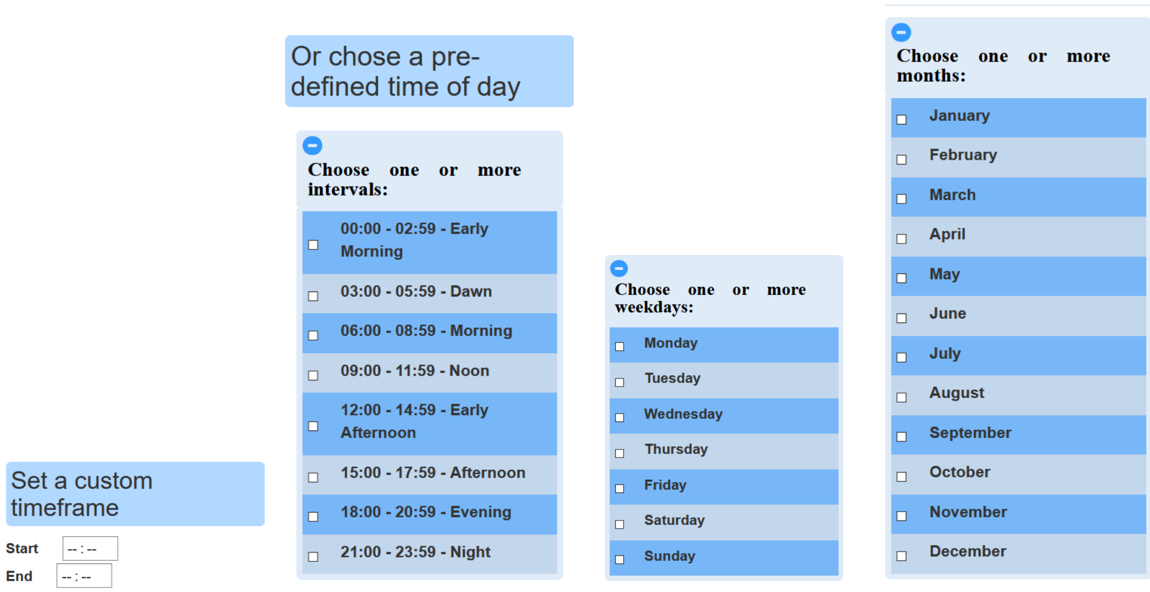
Task	Total time	Sensor / Admin.	Semantic / Visual	Values used	Attrs. used	Results found
LSC 25	0.97	0.10	0.48	5	4	8
LSC 37	0.98	0.09	0.48	4	3	2
LSC 26	0.73	0.20	0.46	5	4	3
LSC 30	0.05	0.00	0.01	4	3	2
LSC 39	1.00	0.21	0.42	5	4	82

3 USER INTERFACE

The client is browser-based, implemented using Javascript, HTML and CSS. First, the user is presented with the main screen, shown in Figure 4 which allows the user to define a compound search query by enabling a variety of filters. Many LSC tasks contain some form of time-related information, such as time of the day or day of the week. Figure 5 shows more details of the time-based filters available to the user. The filter for arbitrary time ranges can be applied only once, but all other time-related filters can have multiple values, which is useful when the tasks include uncertainty.

After selecting the filters and submitting the query, the user can then look at the images returned by the query. An example of this is shown in Figure 6. If the query should return more than 200 images, only the first 200 are shown to avoid long loading times. The user may then choose to refine some filters to narrow down the result.

After the user has narrowed down the pool of potential target images, it is possible to explore the result set in more detail, by clicking on one of the images to view a larger version, along with all the metadata from both MongoDB document collections (Figure 7). This view offers various different ways of viewing the individual image in its temporal context. It is possible to either directly view the few previous/next images, or alternatively enter custom time intervals, for which the system then will retrieve the corresponding images - based around the currently selected image. The latter is illustrated in Figure 8. In tasks involving information like “I had been driving for an hour” or “I had just come from a certain activity” this feature may be highly beneficial.



(a) Custom time intervals. (b) Pre-set time intervals. (c) Selection of weekdays. (d) Selection of months.

Figure 5: Detailed view of the time-related filters. The user can only select one custom time interval (a), but can select multiple values for the other filters.

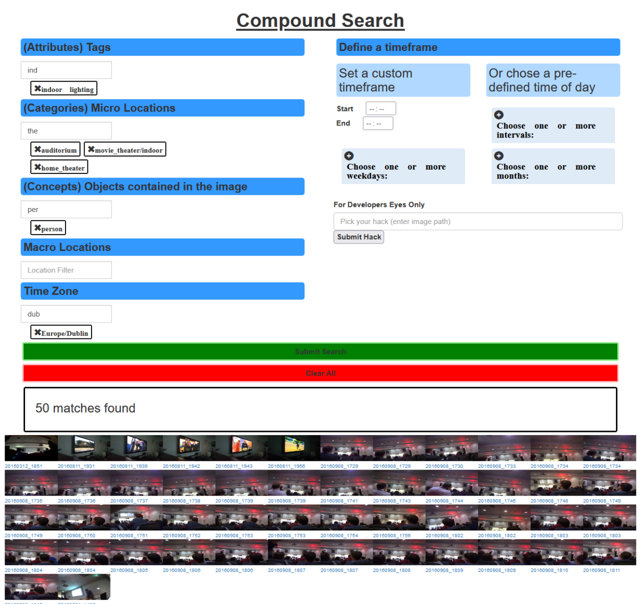


Figure 6: A listing of potential target images, as a result of filling out certain filters and then submitting the preliminary information to the server.

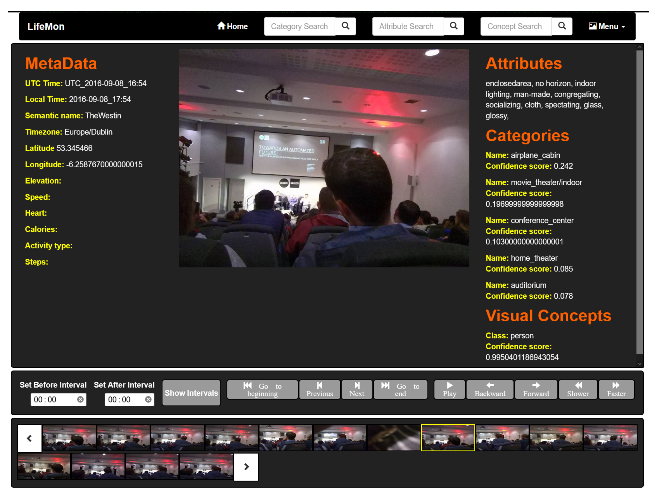


Figure 7: View of a single image, with associated administrative, sensor, semantic & visual metadata. Below it are controls for various kinds of timeline browsing.

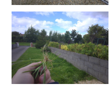
4 EVALUATION

This section describes the results of a preliminary user study, designed to evaluate the potential of LifeMon and to gather feedback for improving the system and its interface.

4.1 Setup

A total of six users participated in the study, 2 females and 4 males. Their education levels ranged from a high-school student to a post-doctoral researcher. Of the four university students, two were IT students. In terms of lifelog experience, one of the IT students participated in the early development of LifeMon and the post-doctoral researcher has participated twice in LSC with his own system, but had never seen LifeMon. All in all, the skill level of the user group covers a very wide range of capabilities.

Table 2: Overview of LSC 2019 tasks used in the user evaluation of LifeMon. In the task description, // indicates the information added every 30 seconds.

Task	Description	Ground Truth
Demonstration Task		
LSC 25	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.	 + 2 more
Sandbox Task		
LSC 37	I remember I was washing clothes. // I think it was white shirts. // Using the clothes washing machine (front loading machine) // in my home. // I recall all the red lights were turned on; perhaps the machine was broken. // It was the weekend, a Saturday.	
Evaluation Tasks		
LSC 26 (expert)	A red car beside a white house // on a cloudy day. // I had driven for over an hour to get here. // It was a Saturday // in August // and it was in the early afternoon.	
LSC 30 (expert)	Pulling up grass or weeds // in my garden // on a cloudy day. // There are trees in my garden // and more trees just outside across the street. // It was a Saturday afternoon.	
LSC 39 (novice)	Watching people speak in a crowded auditorium. // They were talking about 'automated futures'. // It was full of people // and I was at the back of the room. // Afterwards I went for a walk through a historical university campus // on a Thursday evening.	 + 86 more

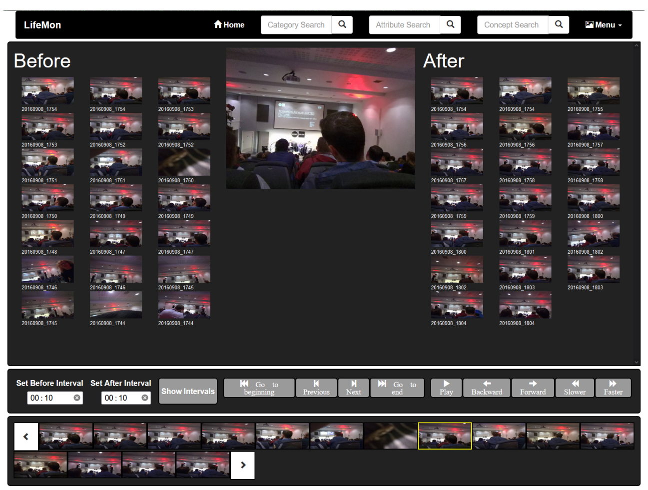


Figure 8: View of a single image with images from before and after on the respective left- and right-hand sides.

We used a set of tasks from LSC 2019 to drive the evaluation. LSC 2019 had a set of 24 distinct tasks, simulating information and data needs that are designed for modelling examples from real life. Each task contains a textual description of an event or a moment in the lifeloggers life. In the competition, the task texts are divided into 6 separate parts that are revealed at 30 second intervals, the first at the start of the task and the last after 150 seconds. The target result of solving each task is a set of images, often one and usually few, but with as many as 94 images in one case. In LSC 2019, a task was considered solved if the user identified one of the correct images as an answer within a time limit of 180 seconds minutes. Unlike the LSC competition, no penalties were assigned to false submissions.

We chose five tasks to use in the user study, as shown in Table 2. The tasks were chosen because (a) they could be solved with LifeMon, and (b) they represented a diverse set in terms of difficulty and number of images in the result set. During the evaluation study, users were first introduced to lifelogging in general and LifeMon in particular. After showing them how to solve the demonstration task, they could practice with the sandbox task, getting help from the experimenter as needed. Finally, they were given the evaluation tasks step by step and asked to solve them. While no hard time limit was set, most tasks were solved within three minutes.

4.2 Results

Table 3 shows the average performance across all users. The first column shows the tasks, while the second column shows how many users completed the individual tasks. As the table shows, two tasks could be solved by all users and one task by four users.

The two failed tasks could not be solved as a result of features lacking in the system. Specifically, it would have been greatly beneficial to be able to use information involving dominant colors and temporal relationships (e.g. "I had been driving for an hour"). Also, better semantic features and query expansion capabilities could make a significant difference; for instance selecting all kinds of houses, or all kinds of gardens, rather than only one specific kind.

The last column of Table 3 shows how many of the task steps the users needed to solve the tasks. Here, the two cases where users did not manage to complete the task are counted as 6 steps. The column shows that for two tasks, users generally needed all the steps to solve the task. The fact that LSC 30, the task with the lowest completion percentage, simultaneously is the one that was solved fastest on average, is interesting. We believe this is because some

Table 3: Task performance across all six users.

Task	Task Completion	Steps Required
LSC 26	6 / 6	5,7 / 6
LSC 30	4 / 6	4,3 / 6
LSC 39	6 / 6	5,7 / 6

users were more creative than others in selecting visual attributes to filter by, which indicates that novice users should be better supported.

Looking at individual users, we observed the lifelogging researcher performed best, with one of the IT students a close second. The other IT student, the former LifeMon developer, solved all tasks, but actually focused more on trying out the system than solving the tasks.

4.3 Discussion

Overall, LifeMon appears a viable LSC system, as all users were able to solve at least two out of three tasks within the given time. That most users needed all six steps of the task to do so, however, is mostly due to the fact that time-related information was usually released in the last one or two steps. In contrast to this, users were often unable to use much of the information released in the first steps to their advantage. Some spent the time until the next step arbitrarily guessing, while others accepted needing more information and waited. The results thus support our initial assumption: the system is technically capable of solving many of the tasks, but having insights into the collection and its structure helps.

The study also indicated some areas in which the system could be improved, notably the following:

(1) The system does not support solving tasks with temporal relationships, such as "I had been driving for an hour." A workaround of using either the timeline or the image bar to browse the hour in either direction is possible, but only the experienced lifelog researcher attempted this.
(2) Several users indicated that colour information would be helpful, as many tasks mention colours. This is of course a complex subject, as the color information often refers to specific parts of the image.
(3) Some users complained about the three different semantic text search boxes, indicating that the front-end should merge these boxes and then have the system identify the appropriate attributes to query by in each case.
(4) Lastly, it became clear that while LifeMon is capable of querying the data provided, that is not always enough, as occasionally users had to improvise semantic relationships between search terms. An example of this is in the demonstration task, where flowers are mentioned but the semantic concepts actually contain a vase. Some form of automatic semantic query expansion could be integrated to address this problem - for instance similarity functions based on the Word2Vec or Glove embedding techniques. Also, we observed that the quality of the existing semantic labels could be improved.

5 CONCLUSION

We have presented LifeMon, a new lifelog retrieval prototype targeting LSC. LifeMon is based around the MongoDB document store, which is one of a host of scalable NoSQL systems developed over the last two decades, with a semi-structured data model that seems well matched with lifelog requirements. We reported on a preliminary evaluation, where six users with a wide range of computing and lifelogging experience, were asked to solve three LSC tasks. The results indicate that even novice users can successfully use LifeMon to solve some LSC tasks, but also indicate some potential areas of improvement.

ACKNOWLEDGMENTS

Christoffer Dyring Heiberg contributed to the development of the LifeMon prototype in the early phases.

REFERENCES

[1] André Calçada and Jorge Bernardino. 2019. Evaluation of Couchbase, CouchDB and MongoDB using OSSpal. In *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*. ScitePress, Vienna, Austria, 427–433.
[2] Aaron Duane, Cathal Gurrin, and Wolfgang Huerst. 2018. Virtual Reality Lifelog Explorer: Lifelog Search Challenge at ACM ICMR 2018. In *Proceedings of the 2018 ACM Workshop on The Lifelog Search Challenge*. ACM, Yokohama, Japan, 20–23.
[3] Aaron Duane, Björn Þór Jónsson, and Cathal Gurrin. 2020. VRLE: Lifelog Interaction Prototype in Virtual Reality: Lifelog Search Challenge at ACM ICMR 2020. In *Proceedings of the Third ACM Workshop on Lifelog Search Challenge, LSC@ICMR 2020*. ACM, Dublin, Ireland, 7–12.
[4] Jim Gemmell, Gordon Bell, Roger Lueder, Steven Drucker, and Curtis Wong. 2002. MyLifeBits: Fulfilling the Memex Vision. In *Proc. ACM Multimedia*. ACM, Juan les Pins, France, 235–238.
[5] Cathal Gurrin, Tu-Khiem Le, Van-Tu Ninh, Duc-Tien Dang-Nguyen, Björn Þór Jónsson, Jakub Lokoč, Wolfgang Hurst, Minh-Triet Tran, and Klaus Schoeffmann. 2020. An Introduction to the Third Annual Lifelog Search Challenge, LSC'20. In *ICMR '20, The 2020 International Conference on Multimedia Retrieval*. ACM, Dublin, Ireland, 584–585.
[6] Cathal Gurrin, Klaus Schoeffmann, Hideo Joho, Bernd Münzer, Rami Albatal, Frank Hopfgartner, Liting Zhou, and Duc-Tien Dang-Nguyen. 2019. A Test Collection for Interactive Lifelog Retrieval. In *Proc. International Conference on MultiMedia Modeling (MMM)*. Springer, Thessaloniki, Greece, 312–324.
[7] Cathal Gurrin, Alan F. Smeaton, and Aiden R. Doherty. 2014. LifeLogging: Personal Big Data. *Foundations and Trends in Information Retrieval* 8, 1 (2014), 1–125.
[8] Cathal Gurrin, Björn Þór Jónsson, Klaus Schöffmann, Duc-Tien Dang-Nguyen, Jakub Lokoč, Minh-Triet Tran, Wolfgang Hürst, Luca Rossetto, and Graham Healy. 2021. Introduction to the Fourth Annual Lifelog Search Challenge, LSC'21. In *Proc. International Conference on Multimedia Retrieval (ICMR'21)*. ACM, Taipei, Taiwan.
[9] Silvan Heller, Mahnaz Amiri Parian, Ralph Gasser, Loris Sauter, and Heiko Schuldt. 2020. Interactive Lifelog Retrieval with vitrivr. In *Proceedings of the Third ACM Workshop on Lifelog Search Challenge, LSC@ICMR 2020, Dublin, Ireland, June 8-11, 2020*. ACM, Dublin, Ireland, 1–6.
[10] Björn Þór Jónsson, Omar Shahbaz Khan, Dennis C. Koelma, Stevan Rudinac, Marcel Worring, and Jan Zahálka. 2020. Exquisitor at the Video Browser Showdown 2020. In *Proceedings of the International Conference on MultiMedia Modeling (MMM)*. Springer, Daejeon, South Korea, 796–802.
[11] Omar Shahbaz Khan, Mathias Dybkjær Larsen, Liam Alex Sonto Poulsen, Björn Þór Jónsson, Jan Zahálka, Stevan Rudinac, Dennis Koelma, and Marcel Worring. 2020. Exquisitor at the Lifelog Search Challenge 2020. In *Proceedings of the Third Annual Workshop on Lifelog Search Challenge*. ACM, Dublin, Ireland, 19–22.
[12] Frantisek Mejzlík, Patrik Veselý, Miroslav Kratochvíl, Tomás Soucek, and Jakub Lokoc. 2020. SOMHunter for Lifelog Search. In *Proceedings of the Third ACM Workshop on Lifelog Search Challenge, LSC@ICMR 2020, Dublin, Ireland, June 8-11, 2020*. ACM, Dublin, Ireland, 73–75.
[13] Abigail J. Sellen and Steve Whittaker. 2010. Beyond Total Capture: A Constructive Critique of Lifelogging. *Commun. ACM* 53, 5 (May 2010), 70–77. https://doi.org/10.1145/1735223.1735243
[14] Ly-Duyen Tran, Manh-Duy Nguyen, Nguyen Thanh Binh, Hyowon Lee, and Cathal Gurrin. 2020. Myscéal: An Experimental Interactive Lifelog Retrieval System for LSC'20. In *Proceedings of the Third Annual Workshop on Lifelog Search Challenge*. Association for Computing Machinery, Dublin, Ireland, 23–28.