Memento: A Prototype Lifelog Search Engine for LSC'21

Naushad Alam
Insight Centre for Data Analytics, Dublin City University
Dublin, Ireland
naushad.alam@insight-centre.org

Yvette Graham
School of Computer Science and Statistics, Trinity College
Dublin, Ireland
ygraham@tcd.ie

Cathal Gurrin
School of Computing, Dublin City University
Dublin, Ireland
cathal.gurrin@dcu.ie

ABSTRACT

In this paper, we introduce a new lifelog retrieval system called Memento that leverages semantic representations of images and textual queries projected into a common latent space to facilitate effective retrieval. It bridges the semantic gap between complex visual scenes/events and user information needs expressed as textual and faceted queries. The system, developed for the 2021 Lifelog Search Challenge, also has a minimalist user interface that includes primary search, temporal search, and visual data filtering components.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; **Search interfaces**; **Retrieval models and ranking**.

KEYWORDS

lifelog, information retrieval, semantic image representation

ACM Reference Format:
Naushad Alam, Yvette Graham, and Cathal Gurrin. 2021. Memento: A Prototype Lifelog Search Engine for LSC'21. 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/10.1145/3463948.3469069

1 INTRODUCTION

Lifelogging can be defined as the process of passively gathering, processing, and reflecting on life experience data collected by an individual using a variety of wearable sensing devices[13]. Lifelogging as a concept has a long history and was initially introduced by Vannevar Bush in 1945. He proposed a hypothetical electromechanical device (the Memex) for the purpose of storing personal information and retrieving it with "exceeding speed and flexibility"[4].

However, lifelogging has evolved into a popular research area in the last couple of decades after the seminal MylifeBits project[10] (Gemmel and Bell). The popularity of lifelogging was further aided by the rapid growth of low-cost sensing devices and enhanced data storage capabilities. Lifelogging has a wide range of application domains, such as memory aids[2], health monitoring[1], activity recognition [5], etc.

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

ACM ISBN 978-1-4503-8533-6/21/08.
https://doi.org/10.1145/3463948.3469069

The Lifelog Search Challenge (LSC) is a comparative benchmarking workshop founded in 2018 [14] to foster advances in multimodal information retrieval similar to previous activities like NTCIR-Lifelog tasks[11] and the ImageCLEF Lifelog tasks[7]. LSC poses a unique information retrieval problem to the participants, where the major challenges are on two fronts, one being efficient data organization and structuring as lifelog data is multimodal, noisy, and repetitive, and the other is regarding the contextual understanding of data to facilitate event retrieval. The evaluation queries in LSC are structured in a way so as to mimic how humans recall events from their daily life, revealing information gradually at a 30 seconds interval and sometimes negating/correcting earlier revealed information.

In this work, we introduce a prototype system that has been developed to participate in the 2021 edition of Lifelog Search challenge. Our system aims to address the challenge of interactive lifelog retrieval on two fronts, bridging the semantic gap between queries and images, and supporting the efficient searching/browsing of the lifelog data. We use the CLIP model [22] to generate semantic representations from the lifelog images which help us better encode the meaning and the relationship among the entities within the image. This is because the model is not optimized for one single task but rather, can perform a variety of them e.g object recognition, activity recognition, scene recognition, optical character recognition etc. Moreover, the model supports instructions in natural language which is similar to how the evaluation queries in LSC are structured allowing us to dictate complex visual scenes efficiently. Our system also has a minimalist user interface that is simple and easy to use. We have integrated visual data filtering (discussed in section 3.7) into our system to help users get an overview of the search results quickly and aid them in better decision-making given the time sensitive nature of the competition.

Memento also supports temporal event search allowing the user to search for some event in close vicinity of another one, as well as temporal navigation to sequentially browse the images around a probable target image. These features were included in the system to tackle those evaluation queries which focus more on temporal information as compared to visual information about the scene/event.

2 RELATED WORK

Since the inaugural Lifelog Search Challenge held in 2018, the event has attracted significant levels of attention and active participation from the research community.

During the last three years, several novel ideas have been proposed to approch this task. A fully immersive virtual reality interface to query lifelog data [9] was proposed at the first event in 2018, which was also the best performing system that year. Several video

retrieval systems [18] [17] [15] which did well in the VBS challenge have also participated in previous LSC events (2018, 2019, 2020) with some modifications/improvements to their original system. The Exquisitor system [16] used relevance feedback to build a model of the user's information needs without using any explicit query mechanism, while SOMHunter[20] and THUIR [19] employed user feedback to iteratively refine the retrieved results. MySceal[24] proposed a temporal query mechanism that allowed to search for up to 3 consecutive events simultaneously and also introduced a concept weighing methodology to determine the importance of visual annotations in the data.

Several systems have tried to address the issues regarding semantic gap between query and images, and poor contextual understanding of the data. FIRST [25] uses an autoencoder like approach to map query text and images into a common semantic space to measure the similarity between them, LifeGraph [23] used a knowledge graph to represent the lifelog data to capture the internal relations of the various data modalities and linked it to external static data sources for better semantic understanding. Chu et al[6] extracted relation graphs from lifelog images to better describe the relationship between entities (subject-object) present within the image.

Our proposed system addresses the semantic gap issue by generating contextual representations for both image and query text using a pretrained model[22] and ranking the images based on cosine similarity scores. Since the model is not optimized for a specific task but can rather perform a variety of tasks such as object recognition, scene recognition, activity recognition, optical character recognition, etc., the generated image representations are semantically rich and encode a lot more information about the scene. This allows the user to retrieve visually complex scenes efficiently using natural language queries.

3 SYSTEM OVERVIEW

In this section, we present an overview of the LSC Dataset and discuss the core components of our system such as semantic image representation, search engine, user interface, etc., in detail, and the enhancements/modifications we did to the existing metadata to further improve it. We also further elaborate on the system's temporal search and navigation functionality and their underlying algorithms.

3.1 LSC Dataset

LSC 2021 dataset [14] is slightly smaller (~ 8K fewer images) as compared to the dataset of last year's challenge which had ~191K images collected using a wearable camera from a single lifelogger spanning 114 days. The data also includes two sets of metadata,

- **Visual Concepts:** This file contains scene descriptions, object tags with confidence scores, object bounding boxes, etc., for each image in the dataset.
- **Biometrics/Location Data:** This dataset contains location, activity, elevation, and biometric data such as calories burnt, heart rate, step count, etc., captured from a wearable device at 60 seconds intervals.

3.2 Semantic Image Representation

The lifelog dataset provided by the organizers is richly annotated where every image is associated with tags for the detected objects, scene descriptions, etc. However, the associated metadata though comprehensive fails to convey the semantic meaning within the image or the relationship between the detected objects. The evaluation topics in LSC, which follow a conversational style however, demand a finer understanding of the visual concepts and relationship among entities within the image.

To facilitate semantic search over lifelog images we use the CLIP model[22] from OpenAI to encode images into high-dimensional representations which captures the overall semantics of the scene. The CLIP model[22] is trained on a large corpus of 400M pairs of images and captions sourced from the internet. The network is not directly optimized for a specific task but is trained on a proxy objective of matching the captions with their respective images. This training objective allows the model to learn a wide variety of visual concepts which can then be used to solve multiple downstream tasks like object and activity detection, optical character recognition, image retrieval, etc., using natural language instructions.

The most crucial aspect of the CLIP model, however, is its zero-shot capabilities on several benchmarks, which allowed us to use the pre-trained model for our use case without any data specific fine tuning. We evaluated the model performance on various metrics (discussed in section 4) and observed that the model is generalizing well on the lifelog data and is able to comprehend finer details and relationships within the image.

3.3 Metadata Enhancement

The metadata provided with the lifelog dataset contains visual concepts/annotations for the images and also has information like location, activity, date/time,etc., which is gathered from a wearable device. Our focus here was to enhance and enrich the specific part of the metadata dealing with location, activity type, date/time, etc., wherever possible as these play a crucial role in information retrieval given the fact that LSC evaluation queries explicitly reveal these bits of information during the search process.

- **Imputing 'semantic name' (Location Name):**
  The location name information is missing at a lot of places in the metadata, hence we tried to impute it wherever the location co-ordinate information was available to us. Location name was also important to us from the event segmentation perspective as our approach (discussed in section 3.4) uses this information to establish event boundaries. We initially created a dictionary of location names (key) and location co-ordinates (value) from the available information. In those cases where one location name had multiple co-ordinates (slightly deviated) we chose the mode value as the final co-ordinate of that location. To impute missing location names, we derived the distance between the co-ordinate for which location name is missing and all co-ordinates from our dictionary to assign a closest possible location name keeping in mind a threshold value of ≤ 3 KMs.
- **Identifying blurred images:** Lifelog images are captured from a wearable camera that takes pictures at regular intervals, no matter the place or activity. This leads to a large

volume of images in the dataset being blurry or occluded and hence not very useful [12]. We tried to identify and tag blurred images in the dataset using an implementation of variance of the laplacian method[21] in the OpenCV[3] library. The objective of this exercise is to minimize less useful images during browsing of temporal events (discussed in section 3.6) as more than 30% of the images in the corpus have some degree of blurring/occlusion.

- **Deriving specific fields from existing data:** We observed from the evaluation topics of previous editions of LSC that having some key data columns like city, hour of the day, name of the day, etc., in our dataset would help us do data filtering effectively. Hence we extracted relevant fields from the given metadata to later use them directly in our faceted filtering functionality.

3.4 Segmentation of Each Day into Events

It has been known for a long time that Lifelog data is inherently sequential in nature where each day can be broken into coherent and meaningful chunks called 'events'. E.g. *driving in the car from home to the office* can be one event while *walking from office to the cafeteria to grab a coffee* can be another. We define event boundaries based on how the current activity and location of the lifelogger changes compared to what they were at a previous time step. We devised a rule-based algorithm that evaluates the difference in data (location name and activity) sequentially to determine an event change, and assigns an event number which is a unique ID for each event. However, there are edge cases that we have carefully considered. E.g. when the lifelogger exits his/her home and gets into his/her car the location name and activity might change from (Home - None) to (None-Transport) indicating a change in the event, however when driving across the city the location name will continuously change while activity will remain static. In the latter scenario the algorithm handles it as a single event despite changes in location until the lifelogger finishes the car driving activity.

Previous approaches [8] [19] [24] to event segmentation in a lifelog context have used image similarity metrics to decide the event boundaries. i.e. images are processed chronologically and their similarity is calculated with neighboring images (vector similarity), where two dissimilar neighbors indicate the start of a new event. We however approach event segmentation based on user activity data as our goal is to do data filtering based on the previous, next and current activity of the user as LSC queries also explicitly specify user activity in temporal queries e.g *Walking on a green footpath, to my car. I got into my car and drove to a lunch* which indicates the current activity as walking and the next activity as driving.

3.5 Search Engine

The image search and ranking functionality of our proposed system, Memento is built using the Flask framework. The functionality is delivered to the end-user via RESTful APIs that outputs results in JSON format. Figure 1 shows a high-level overview of the system architecture.

Following is the sequence of execution from query to end result:

(1) End-user sends query string to the server using the web client.
(2) The query string is translated to its vector representation using the text encoder of the CLIP model.
(3) The image representations are already on the server as a static npy file. The query vector is compared with image representations using cosine similarity which gives us a ranked list of image indices.
(4) Image metadata is then fetched from a static CSV file stored on the server using the image indices and the top 2000 are returned to the web client as a JSON response. The quality of the retrieved results will depend on how well the query was 'engineered'. Since the model is trained on image-caption pairs, the query should mimic the language style of image captions to get better results which the authors of [22] call 'prompt engineering'. LSC queries usually have information scattered across multiple sentences which should be rewritten in a compact way before a query can be initiated.

The design to store image features and metadata as static files is by choice, to leverage the power of vectorized operations for a much faster turnaround time.

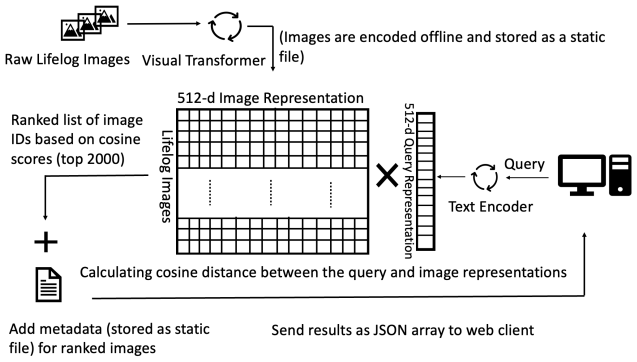

Figure 1: System Architecture

3.6 Temporal Search and Navigation

Some of the LSC evaluation queries do not describe the visual scene in great detail but they rather focus more on specifying temporal events surrounding the main event. For e.g.

- Main Event: Buying a ticket for a train in Ireland from a ticket vending machine.
- Next Event: After the purchase, I walked up stairs to the platform and waited 8 minutes for the train to arrive.
- Previous Event: I had walked (for 36 minutes) to the station after eating sushi and beer.

The temporal search functionality hence allows the user to efficiently search such queries by specifying main and temporal events (previous and next) as separate inputs to the system. Temporal search in our proposed system is inspired by [24]. Our approach however is fundamentally different as we leverage semantic representations to search for temporal events similar to how searching works across the system. We have also defined our search space using 'event numbers' as opposed to time duration because we felt

it would help reduce the noise in our final result set. The temporal search functionality of our system has the following execution steps:

(1) The user initiates a query to search for the main event (similarly as discussed in section 3.5).
(2) Once the user has the ranked result set (based on cosine similarity scores) on the screen, a temporal search can be initiated by specifying either a previous event or next event, or both.
(3) The temporal search algorithm iterates through the initial result set to search for previous and next events in a predefined search space, which is (current event -2) for the previous and (current event + 2) for the next event. (Every image in our dataset has an event number associated to it already as discussed in section 3.4).
(4) The algorithm assigns temporal scores (previous and next) to every image in the initial result set, which is the maximum cosine similarity score within their respective search spaces.
(5) The final score of each image is then computed as the sum of temporal scores and initial cosine similarity score based on which the images are re-ranked and rendered on screen.

The efficacy of this algorithm depends on how well the system locates and ranks the main event. Searching for temporal events when the main event is not within our initial result set, is futile. However, we show in section 4 that there is a very high probability of the target image being in the top-2000 results given that the query string is engineered well.

The system also supports sequential browsing of previous and next non-blurred images around a probable target image as in some scenarios browsing is fast and sufficient to arrive at a decision.

3.7 User Interface

The user interface was designed with the goal of developing a clutter-free as well as feature-loaded system. We have tried to maximize the result set visibility by separating the functionalities like data filtering, temporal search, image starring, etc., into separate overlay windows which helps to minimize clutter.

- **Primary Search Interface:** This interface has two major components, one is the primary navbar on top of the window which embeds the search box and buttons to access system functionalities while the other one is the component to display search results. Figure 2 shows a snapshot of the primary search interface.
- **Data Filtering Component:** The goal of this interface is to provide data filtering functionality and at the same time convey a mental picture of the data to the user to aid better decision making. This component allows the users to filter the result set on the basis of day, city, time, year, month and activities or any combinations of these. Figure 3 shows a snapshot of the data filtering interface.
- **Starred Images:** This functionality allows the user to bookmark/star a particular image from the result set to view it later. It displays the starred images as well as relevant metadata associated with it. The user can choose to submit the image to the evaluation server using the 'Submit' button or initiate temporal browsing to view previous and next images in sequence using 'Inspect' functionality. Figure 4 shows a snapshot of Image starring interface rendered in an overlay window.

4 SYSTEM EVALUATION

We evaluated our proposed system on the 24 evaluation topics from LSC 2019. The evaluation topics reveal information sequentially in parts at a regular time interval of 30 seconds (starting at t=0s up to t= 150s), usually giving out visually descriptive information early on and more explicit information like time, date, place, etc., at later stages. We chose to evaluate our search engine by only considering the visual information available to us by t= 60 seconds because the back end model powering the search engine has no sense of explicit information like time, date, etc. The evaluation was performed automatically; however, the evaluation queries going in as input to the backend model were tweaked manually to make them more compact.

We evaluated our system on the following metrics:

(1) Hit@K: For a given topic, Hit@K is defined as finding at least one target image among top-K images in the result set;
(2) Precision@K;
(3) Recall@k.

t	@1	@3	@5	@10	@20	@50
0s	8.33	25.00	29.17	29.17	37.50	50.00
30s	8.33	25.00	25.00	33.33	33.33	54.17
60s	12.50	29.17	29.17	41.67	54.17	75.00

Table 1: Hit@K calculated at different amounts of elapsed times, t and K values across 24 evaluation topics for LSC'19

Table 1 shows the hit percentages calculated for 24 evaluation topics from LSC 2019 at different values of K and t. At t=60s and K=50, we are able to find at least one target image in top-50 results for 75% of the evaluation topics (18 out of 24 topics).

Figure 5 shows variation in hit percentage as we scale up K. We observe a rapid increase in hit percentage as K increases up to 50, after which the curve almost flattens outs at a maximum of 87.5% (21 out of 24 topics) within the 1-2000 range for K. These results provide evidence on the efficacy of the search engine given the fact that we are only using a portion of the information (up to t=60s) for each topic.

We further evaluated our system on precision and recall metrics at multiple K values. Figure 6 shows the precision versus recall curve at K = 1 to 100 averaged across 24 evaluation topics from LSC 2019 with only considering the information available to us by t=60s.

We observe an averaged maximum recall of 36.2% at K =100 while average maximum precision recorded is 15.2% at K=3. These numbers however are relatively poor as compared to the system's performance on Hit@K metric discussed above. The reasons for this could be the way LSC queries and their respective ground truth images are structured. LSC queries typically search for some specific event in time and they initially yield a visual description of the scene/event. However, there might several images in the lifelog

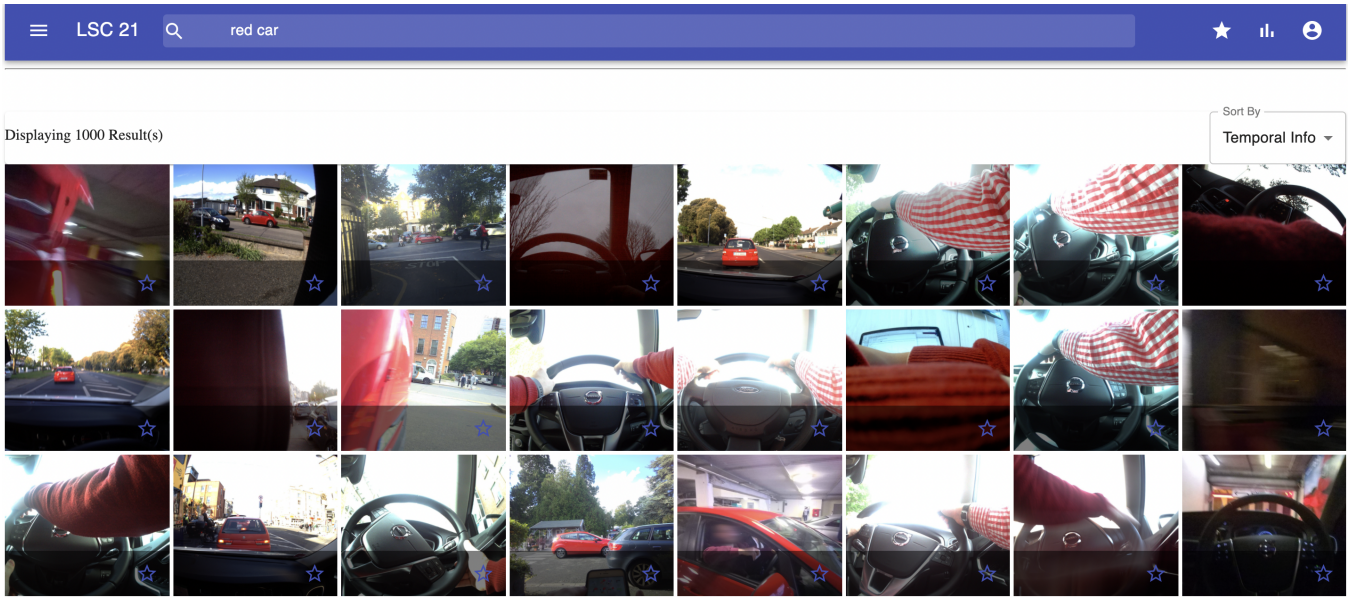

Figure 2: Primary Search Interface

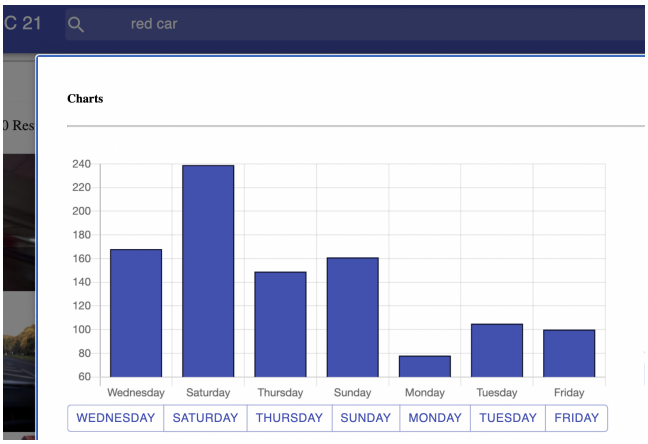

Figure 3: Visual data filtering interface displaying day filters rendered in an overlay window. Visual filters for city, time, year and activities (previous-current-next) are also displayed in the same window.

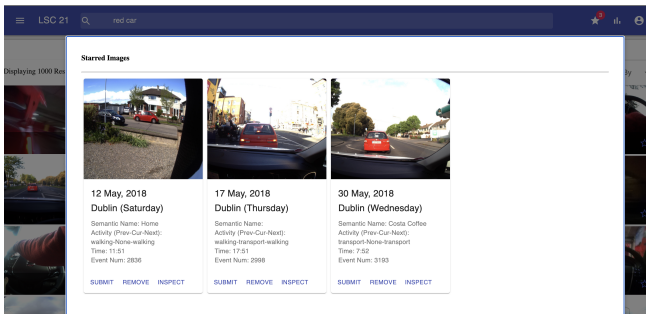

Figure 4: Interface for starred images rendered in an overlay window

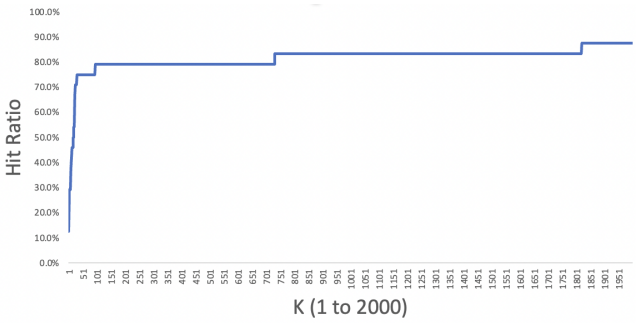

Figure 5: Hit percentage plotted against different K values (1 to 2000) considering information available at t=60s

dataset that match the description but are unrelated to the actual event, such as *a red car on a cloudy day, looking at flowers and lamp* and so on. In these scenarios, the search engine, despite correctly finding images with the given visual description, might not be able to perform well on precision as the ground truth images belong to one specific day, time or place.

Furthermore, in the case of queries where temporal events are also specified along with the main event, the ground truth usually consists of images from all events (temporal and main). For example, for a query *Watching people speak in a crowded auditorium. Afterwards I went for a walk through a historical university campus,* the ground truth will contain images of the auditorium as well as of the university campus. In this scenario, if we search for the main event the search engine will never be able to find relevant images

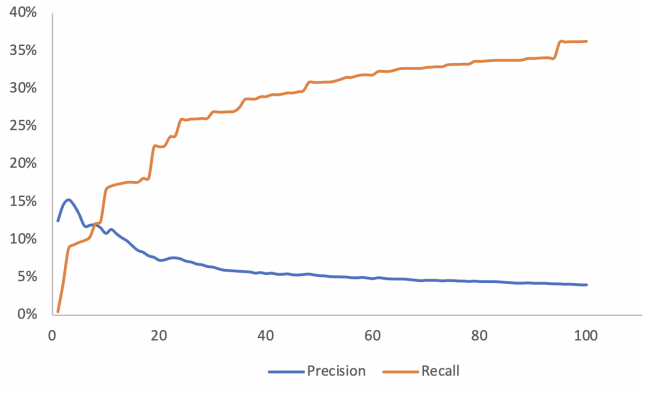

Figure 6: Precision versus Recall curve at K = 1 to 100 averaged across 24 evaluation topics from LSC'19

of the temporal event and vice versa, and hence will perform poorly in terms of recall.

5 CONCLUSION AND FUTURE WORK

In this work, we introduced our lifelog retrieval system called Memento which uses a pre-trained model to generate semantic representations for images and queries. We evaluated our system on multiple metrics and got good results despite testing in a constrained environment which indicates that our system is successful in bridging the existing semantic gap between the two data modalities to a much larger extent.

Since the system accepts natural language queries, the logical next step would be to explore the feasibility of an interactive dialogue based retrieval system.

6 ACKNOWLEDGEMENTS

This publication has emanated from research supported by Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289_P2, co-funded by the European Regional Development Fund.

REFERENCES

[1] Muhammad Bilal Amin, Oresti Banos, Wajahat Ali Khan, Hafiz Syed Muhammad Bilal, Jinhyuk Gong, Dinh-Mao Bui, Soung Ho Cho, Shujaat Hussain, Taqdir Ali, Usman Akhtar, Tae Choong Chung, and Sungyoung Lee. 2016. On Curating Multimodal Sensory Data for Health and Wellness Platforms. *Sensors (Basel, Switzerland)* 16, 7 (June 2016). https://doi.org/10.3390/s16070980

[2] Seyed Ali Bahrainian and Fabio Crestani. 2018. Augmentation of Human Memory: Anticipating Topics that Continue in the Next Meeting. (2018), 10.

[3] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[4] Vannevar Bush. 1945. As We May Think. https://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/ Section: Technology.

[5] Alejandro Cartas, Juan Marín, Petia Radeva, and Mariella Dimiccoli. 2017. Recognizing Activities of Daily Living from Egocentric Images. *arXiv:1704.04097 [cs]* (April 2017). http://arxiv.org/abs/1704.04097 arXiv: 1704.04097.

[6] Tai-Te Chu, Chia-Chun Chang, An-Zi Yen, Hen-Hsen Huang, and Hsin-Hsi Chen. 2020. Multimodal Retrieval through Relations between Subjects and Objects in Lifelog Images. In *Proceedings of the Third Annual Workshop on Lifelog Search Challenge*. ACM, Dublin Ireland, 51–55. https://doi.org/10.1145/3379172.3391723

[7] Duc-Tien Dang-Nguyen, Luca Piras, Michael Riegler, Liting Zhou, Mathias Lux, Minh-Triet Tran, Tu-Khiem Le, Van-Tu Ninh, and Cathal Gurrin. [n.d.]. Overview of ImageCLEFlifelog 2019: Solve My Life Puzzle and Lifelog Moment Retrieval. ([n. d.]), 17.

[8] A. R. Doherty and A. F. Smeaton. 2008. Automatically Segmenting LifeLog Data into Events. In *2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services*. 20–23. https://doi.org/10.1109/WIAMIS.2008.32 ISSN: 2158-5881.

[9] 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. https://doi.org/10.1145/3210539.3210544

[10] Jim Gemmell, Gordon Bell, and Roger Lueder. 2006. MyLifeBits: a personal database for everything. *Commun. ACM* 49, 1 (Jan. 2006), 88–95. https://doi.org/10.1145/1107458.1107460

[11] Cathal Gurrin, Hideo Joho, Frank Hopfgartner, Liting Zhou, Van-Tu Ninh, Tu-Khiem Le, Rami Albatal, Duc-Tien Dang-Nguyen, and Graham Healy. 2019. Overview of the NTCIR-14 Lifelog-3 Task. (2019), 13.

[12] Cathal Gurrin, Alan F. Smeaton, Daragh Byrne, Neil O'Hare, Gareth J. F. Jones, and Noel O'Connor. 2008. An Examination of a Large Visual Lifelog. In *Information Retrieval Technology*, Hang Li, Ting Liu, Wei-Ying Ma, Tetsuya Sakai, Kam-Fai Wong, and Guodong Zhou (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 537–542.

[13] 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. https://doi.org/10.1561/1500000033

[14] 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.

[15] Silvan Heller, Mahnaz Amiri Parian, Ralph Gasser, Loris Sauter, and Heiko Schuldt. 2020. Interactive Lifelog Retrieval with vitrivr. In *Proceedings of the Third Annual Workshop on Lifelog Search Challenge*. ACM, Dublin Ireland, 1–6. https://doi.org/10.1145/3379172.3391715

[16] 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. https://doi.org/10.1145/3379172.3391718

[17] Gregor Kovalčík, Vít Škrhak, Tomáš Souček, and Jakub Lokoč. 2020. VIRET Tool with Advanced Visual Browsing and Feedback. In *Proceedings of the Third Annual Workshop on Lifelog Search Challenge*. ACM, Dublin Ireland, 63–66. https://doi.org/10.1145/3379172.3391725

[18] Andreas Leibetseder and Klaus Schoeffmann. 2020. lifeXplore at the Lifelog Search Challenge 2020. In *Proceedings of the Third Annual Workshop on Lifelog Search Challenge*. ACM, Dublin Ireland, 37–42. https://doi.org/10.1145/3379172.3391721

[19] 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 Lifelog Search Challenge*. ACM, Dublin Ireland, 29–35. https://doi.org/10.1145/3379172.3391720

[20] František Mejzlík, Patrik Veselý, Miroslav Kratochvíl, Tomáš Souček, and Jakub Lokoč. 2020. SOMHunter for Lifelog Search. In *Proceedings of the Third Annual Workshop on Lifelog Search Challenge*. ACM, Dublin Ireland, 73–75. https://doi.org/10.1145/3379172.3391727

[21] J. L. Pech-Pacheco, G. Cristobal, J. Chamorro-Martinez, and J. Fernandez-Valdivia. 2000. Diatom autofocusing in brightfield microscopy: a comparative study. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, Vol. 3. 314–317 vol.3. https://doi.org/10.1109/ICPR.2000.903548

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. *arXiv:2103.00020 [cs]* (Feb. 2021). http://arxiv.org/abs/2103.00020 arXiv: 2103.00020.

[23] Luca Rossetto, Matthias Baumgartner, Narges Ashena, Florian Ruosch, Romana Pernischová, and Abraham Bernstein. 2020. LifeGraph: A Knowledge Graph for Lifelogs. In *Proceedings of the Third Annual Workshop on Lifelog Search Challenge*. ACM, Dublin Ireland, 13–17. https://doi.org/10.1145/3379172.3391717

[24] 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*. ACM, Dublin Ireland, 23–28. https://doi.org/10.1145/3379172.3391719

[25] Minh-Triet Tran, Thanh-An Nguyen, Quoc-Cuong Tran, Mai-Khiem Tran, Khanh Nguyen, Van-Tu Ninh, Tu-Khiem Le, Hoang-Phuc Trang-Trung, Hoang-Anh Le, Hai-Dang Nguyen, Trong-Le Do, Viet-Khoa Vo-Ho, and Cathal Gurrin. 2020. FIRST - Flexible Interactive Retrieval SysTem for Visual Lifelog Exploration at LSC 2020. In *Proceedings of the Third Annual Workshop on Lifelog Search Challenge*. ACM, Dublin Ireland, 67–72. https://doi.org/10.1145/3379172.3391726