MyEachtraX: Lifelog Question Answering on Mobile

Ly-Duyen Tran
lyduyen.tran@dcu.ie
Dublin City University
Dublin, Ireland

Thanh-Binh Nguyen
Vietnam National University Ho Chi Minh City — Ho Chi Minh City University of Science
Ho Chi Minh City, Vietnam

Cathal Gurrin
Dublin City University
Dublin, Ireland

Liting Zhou
Dublin City University
Dublin, Ireland

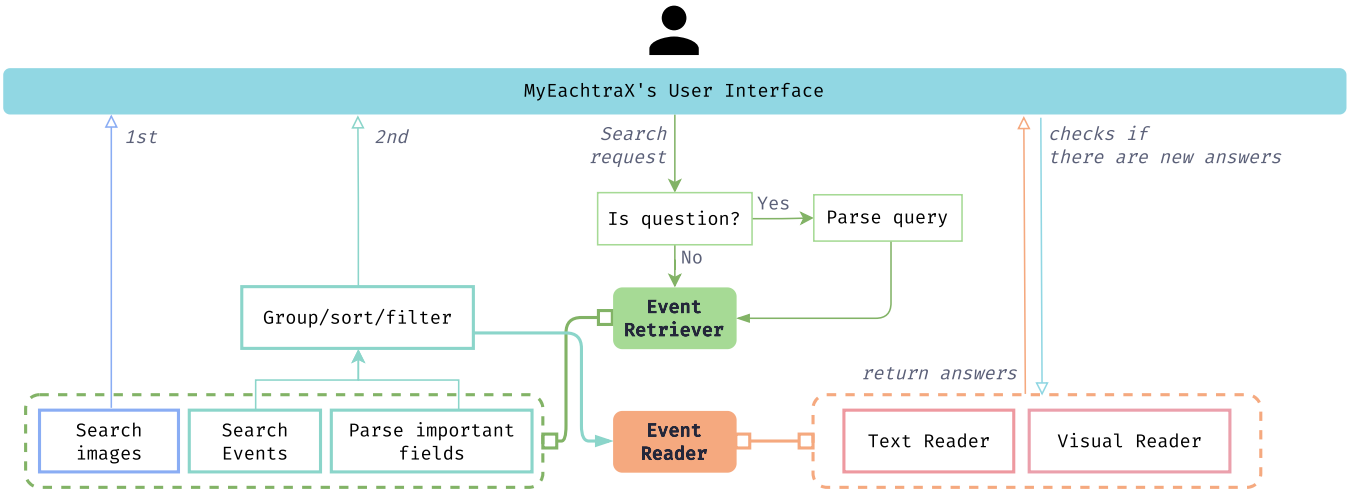

Figure 1: System Design of MyEachtraX. The backend system constantly interacts with the user through the user interface (UI) at every step. Two main components are the Event Retriever and Event Reader, which consist of several asynchronised child components (in the dashed boxes). The lines with arrows indicate the flow of data between the components, while those with squares indicate zooming in on the components.

ABSTRACT

Your whole life in your pocket. That is the premise of lifelogging, a technology that captures and stores every moment of your life in digital form. Built on top of MyEachtra and the lifelog question-answering pipeline, MyEachtraX is a mobile-based application that addresses the overlook of mobile platforms in the area. Furthermore, leveraging the latest advancements in natural language processing, such as Large Language Models (LLMs) and Multimodal Large Language Models (MLLMs), the system enhances the query-parsing, post-processing, and question-answering processes in lifelog retrieval. Official lifelog questions from the previous Lifelog Search Challenges were used to evaluate the system, which achieved an accuracy of 72.2%. We identify the retrieval component as the main bottleneck of the pipeline and propose future works to improve the system.

CCS CONCEPTS

• **Information systems** → Query reformulation; *Presentation of retrieval results*; **Language models**; **Question answering**; **Retrieval on mobile devices**; **Image search**.

KEYWORDS

Lifelog, Mobile, Retrieval, Question Answering

ACM Reference Format:
Ly-Duyen Tran, Thanh-Binh Nguyen, Cathal Gurrin, and Liting Zhou. 2024. MyEachtraX: Lifelog Question Answering on Mobile. In *The 7th Annual ACM Lifelog Search Challenge (LSC '24), June 10, 2024, Phuket, Thailand.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3643489.3661128

1 INTRODUCTION

Imagine having access to your entire life in your pocket. Every moment, every event, every memory, all at your fingertips. This is the promise of lifelogging, a technology that captures and stores every moment of your life in digital form [8]. Lifelogging has the potential

LSC '24, June 10, 2024, Phuket, Thailand

ACM ISBN 979-8-4007-0550-2/24/06
https://doi.org/10.1145/3643489.3661128

to revolutionise the way we manage and navigate our memories, but it also presents a number of challenges, such as how to search and retrieve lifelog data in an efficient and user-friendly way. Existing lifelog search often focuses on image analysis, which can be challenging when it comes to understanding the context of the actual event. For example, Question Answering (QA) remains a challenging task in the field of lifelog retrieval, as it requires more complex reasoning over multiple sources of information. In this work, we leverage the latest advancements in natural language processing, such as Large Language Models (LLMs) and Multimodal Large Language Models (MLLMs), to improve different components of the lifelog retrieval systems. Query parsing, dynamic post-processing, and generative event readers are the main components that are upgraded in this work.

Another challenge is the accessibility of lifelogging technology. Existing lifelog retrieval systems are often designed to be used on desktop computers and overlook the mobile platforms integral to our daily lives. Mobile devices offer several advantages over desktop computers, such as portability, convenience, and always-on connectivity. Personalisation and context-awareness are also more feasible on mobile devices, as they can leverage the sensors and other capabilities of the device. Therefore, for LSC'24 this year [10], we propose a mobile-based lifelog retrieval system that is designed to increase the accessibility and usability of lifelogging technology. The system is based on our previous work of MyEachtra [26], which was the runner-up in the Lifelog Search Challenge 2023 (LSC'23). Our contributions are as follows: (i) MyEachtra's adaptation to a mobile-friendly web app, (ii) a novel integration of generative models for dynamic query-parsing and post-processing, and (iii) a qualitative evaluation of the LLMs' performance in lifelog QA.

2 RELATED WORK

2.1 Lifelog Question Answering

To unlock lifelogging technology's full potential, we must be able to search and retrieve lifelog data. Lifelog Search Challenges (LSCs) have been proposed since 2018 to address such challenges and provide a platform for researchers to develop and evaluate lifelog retrieval systems. Challenges are regularly identified and tackled in the LSCs such as the need for multimodal retrieval, temporal-related queries [25], user-friendly interfaces [12], and QA [24]. The involvement of regular research teams and the public in the LSCs has led to the development of a number of lifelog retrieval systems, which have been shown to achieve state-of-the-art performance on a wide range of lifelog retrieval tasks. Recent developments in the field of natural language processing, especially the development of large language models (LLMs), have also led to interesting potentials for lifelog retrieval systems. An example of such a system is FIRST [11], which uses an LLM to suggest search queries to the user, with a focus on users who are not native English speakers. Another example is LifeInsight [17], where the LLM is presented as a chatbot and supports various tasks, namely, query reformulation. This work is inspired by these systems, emphasising lifelog QA.

Lifelog Question Answering (LLQA) is a relatively new task in the field of lifelog retrieval, which was first introduced in LSC'22 [9] and originally proposed by Tran et al. [21]. In this task, the system is required to answer natural language questions about the lifelog data, for example, *'What did I eat for breakfast on Monday?'*. The task requires more complex reasoning and understanding of the context of the query and the lifelog data, compared to traditional lifelog retrieval tasks. Built on top of MyEachtra [26], a lifelog question-answering pipeline was proposed in [24] to incorporate QA into existing lifelog retrieval systems. The pipeline was inspired by the success of the open-domain question-answering pipeline [4] in the field of natural language processing. A similar trend of Retrieval Augmented Generation [15], a.k.a. RAG, also has a similar architecture to the pipeline, demonstrating the effectiveness of the pipeline in various tasks. The idea of combining retrieval (Event Retriever in LLQA) and generation (Event Reader) will be further explored in this work.

2.2 Lifelog Interfaces

Another crucial aspect of lifelog retrieval is the user interaction with the system. Most systems in the LSCs are designed to be used on desktop computers, with a few exceptions, such as virtual reality [7, 20], speech-based [2], and mobile [13] interfaces. Mobile devices are now ubiquitous and have become an integral part of our daily lives. They offer several advantages over desktop computers, such as portability, convenience, and always-on connectivity. This represents an overlooked research area in lifelog retrieval, where the focus has been on improving the performance of the retrieval algorithms, rather than the user experience. Therefore, consistent with the goal of making lifelog more accessible to the public of our previous works [22, 26], we propose a mobile-based web application for lifelog retrieval.

2.3 Large Language Models

Perhaps the most significant advancement in the field of natural language processing in recent years is the development of large language models (LLMs) [28]. Stemmed for the success of pre-trained Transformer-based [27] models such as BERT [5], LLMs extend this idea of pre-training to a much larger scale, hence the name. These models have been shown to achieve unprecedented performance on a wide range of natural language processing tasks even on few-shot settings, compared to traditional fine-tuning approaches [3]. Kojima et al. [14] investigated the zero-shot reasoning capabilities of LLMs, suggesting enormous knowledge stored in the models on high-level, multitask reasoning. Multimodal Large Language Models (MLLMs) are an extension of LLMs that can process both text and images. MLLMs have been shown to be capable of generating image narratives and answering image-based questions while having a human-like response [19]. Some examples of such models are ChatGPT-4 [1] from OPENAI, Monkey [16], and InternLM-XComposer2 [6], just to name a few. In this work, we leverage the power of both an LLM and an MLLM to improve the performance of the lifelog retrieval system. *Specifically, for the rest of the paper, we refer to ChatGPT-4 as the LLM, while InternLM-XComposer2 as the MLLM. The choice of these models is due to their availability, and realistically, any other models of the same kind can be used.*

3 SYSTEM DESIGN

Following the Lifelog Question Answering pipeline in [24], we focus on two main components of the system: Event Retrieval and

Event Reader. Given a question, the Event Retrieval component retrieves a list of events (and their corresponding information) that are relevant to the question. After that, the Event Reader component can read the retrieved information to find the answer to the question. The advantage of this pipeline is that it can be easily adapted to any search engine that can retrieve lifelog data, in other words, the participating systems in this Lifelog Search Challenge. In this work, we do not focus on the retrieval algorithm but rather extend the pipeline with additional processing steps to improve the performance of the system. As for the Event Reader, we follow the previous approach in [24] to combine a text-based reader and a visual reader in order to provide a more comprehensive answer to the question. Both readers are upgraded using Large Language Models (LLMs) and Multimodal Large Language Models (MLLMs) to provide more informative answers to the user.

The general system design is shown in Figure 1. The next sections will describe the implementation of the system and the user flow of the system.

3.1 Query Parsing

This step is specific to the lifelog question-answering task, where the system is required to answer natural language questions about the lifelog data. First, a rule-based question detector is used to distinguish between known-item search, ad-hoc search, and QA. If the query is a question, we utilise a pre-trained LLM to generate a plain statement, which is descriptive and contains all the necessary information to retrieve the relevant events, similar to a known-item search query. For example, the question *'What did I eat for breakfast on Monday?'* can be converted to *'I am eating breakfast on Monday'*. The prompt for the LLM is a few-shot learning prompt containing several such examples.

3.2 Event Retriever

Event-based retrieval is an approach that was proposed by MyEachtra [26], which defines the retrieval unit as *events*, instead of the conventional image-based retrieval. An event is a collection of images and text that are related to a specific activity or a specific time period. In this approach, each event is indexed as a whole and the ranking algorithm will not take into account the individual images or text in the event. This approach has the advantage of providing a more comprehensive view of the event, which can be useful for the Event Reader component. Another advantage is that the search space can be reduced significantly, especially for long-lasting, continuous, or repetitive activities such as watching the television or working in the office. However, small details in the event can be lost in this approach, which can be a disadvantage for some questions.

To address this issue, MyEachtraX offers *both* approaches and lets the user choose which one they find more comfortable. The user can switch between the two approaches by simply navigating to a different tab on the search result page. The two algorithms are as follows:

- **Image-based retrieval**: This approach is the default approach in MyEachtraX, which has great performance in general. CLIP[18], an image-text model, is used to embed lifelog images into a high-dimensional space. Lifelog queries are also embedded into the same space and the search results are ranked based on the similarity between the query and the images. The search results are displayed as a list of images.
- **Event-based-approach**: This requires segmenting the lifelog data into events. After that, CLIP models are also used to embed lifelog images, whose embeddings are then fed into an aggregation Transformer to generate a single representation of the event. At the retrieval time, the query is embedded into the same space, and the search results are ranked in the same way as the image-based retrieval.

Post-processing is another component that we added to the Event Retriever. In previous work, we forced the Event Reader to produce answers for each event in the top-k search results. However, in many cases, even if the real answer is suggested, other irrelevant answers can push the real answer down the list. Another factor to consider is that low-quality answers might cause the user to lose interest in using the QA feature of the system. Instead, they would prefer to look at the retrieved events and find the answers by themselves.

Ideally, an expert user who understands the system can define what type of data is relevant to the query. However, in practice, novice users might not know what to look for, and it would be unreasonable to ask them to define the filtering criteria, especially under time constraints. Therefore, we employed an LLM to generate the criteria, and allow the user to adjust them if necessary. Specifically, the LLM is asked to generate a list of relevant fields (e.g. location name, weekday, and time of day), the grouping, and sorting criteria (e.g. grouped by day, sorted by time in descending order). For example, for the question *'How many days did I stay in Australia?'*, the LLM might suggest the relevant fields as the country name and date, grouped by day (to count the number of days), and sorted by date in any order.

3.3 Event Reader

To answer the question, the Event Reader would take the processed events from the Event Retriever as input. As mentioned earlier, two different readers are used to address the multimodality of lifelog data. Text Reader is faster to run, but is blind to the visual information in the event. Visual Reader, on the other hand, can provide more accurate answers but is much slower to run. For such reasons, we propose a hybrid approach as follows.

A **Text reader** is based on the same LLM mentioned in the last section. For each event, following the prompting approach of MyEachtra, a rule-based text summarisation is generated with the relevant information. The summarisation of the top-k events is then concatenated and fed into the LLM to generate the answer.

On the other hand, a **Visual Reader** is based on the MLLM model, which can process both text and images. The model is used to embed the images in the event and generate the answer. However, the model is designed for the Visual Question Answering task, which only takes a single image as input. To adapt the model to the lifelog data, we propose using the (global) event embeddings as the input to the model instead of the individual images. In this way, we can speed up the inference time significantly.

All the answers from the two readers are shown to the user when they are ready. The user can choose which answer they find more reliable or more informative.

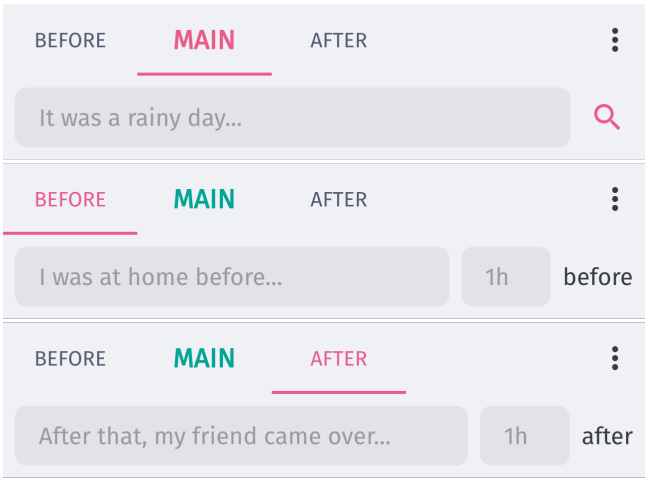

Figure 2: Search bars.

3.4 User Flow

Here we describe the user flow of the system. There are no major differences in different types of retrieval tasks (known-item search, ad-hoc search, or QA) in MyEachtraX. The user flow is designed to be as adjustable as possible and not to force the user to follow a specific path. All the features can be ignored if the user is not interested in them. In this way, we hope to provide a more flexible and user-friendly system. The user flow is as follows:

3.4.1 Enter the query. The user enters the query in the search bar in a natural language form. Temporal queries, such as *'I was home before going to the office. After that, I went to have lunch with my friend'*, which is also supported. The user can enter each part of the query separately by switching the tab above the search bar, as shown in Figure 2. This design is inspired by the search engine in the MyScéal system [23], without sacrificing the space needed for the search bar.

3.4.2 Receive the search results. As stated in the Event Retriever section, the user can switch between the two retrieval approaches by navigating to a different tab on the search result page. At first, they would be shown the search results from the image-based retrieval, which is the default approach and offers a good performance in general. The processed events are shown on another page and will be notified to the user when they are ready. Normally both would be ready at the same time, but in some cases, the post-processing step might take longer to run.

3.4.3 Adjust the filtering criteria. The user can access the advanced search options on the top right of the screen to adjust the filtering criteria generated by the LLM and adjust them if necessary. We expect the user to adjust the criteria only if they are an expert user, or if they are not satisfied with the results. In the latter case, the user can also report the issue to the system, which will be used to improve the system in the future.

3.4.4 Read the answers. If it is a question-answering task, answers will appear as soon as they are ready in the form of phone notifications. The user can click on the notification to see the answers.

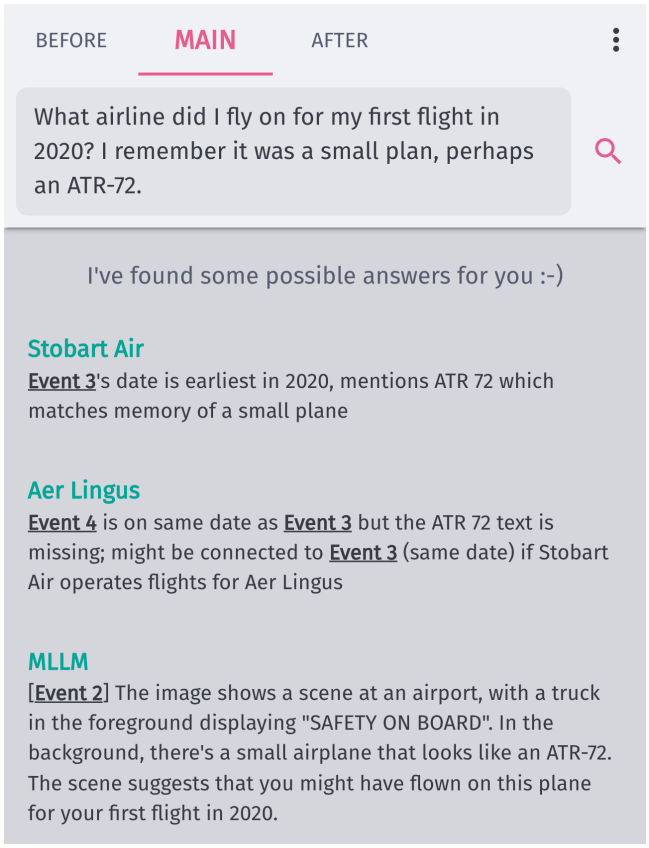

Figure 3: Some sample answers from the system. The first ones were generated by the Text Reader, while the ones with 'MLLM' were generated by the Visual Reader.

The answers, as we can see in Figure 3, are shown in another tab at the bottom. We aim to make the answers as informative as possible, with the relevant events highlighted and can be accessed by clicking on them. However, the length of the answers might need to be limited in order not to overwhelm the user.

3.4.5 Explore the timeline. The user can also explore the timeline of the lifelog dataset, by tapping on any image in the search results. The timeline will appear in the Timeline Page with the events before and after the selected event. The user can also navigate back and forth in the timeline in order to see more events.

To address the issues of limited screen space, we group similar images in the timeline, with a representative image shown for each group. The user can click on an 'Expand' button to see all the images in the group if they are interested. This is illustrated in Figure 4. The pop-up span will close automatically if the user clicks on the background or does not interact with it for a certain amount of time.

3.4.6 Image Viewer. The Image Viewer is a pop-up feature designed to enhance the user experience by displaying images in full-screen mode. It activates when the user either double-taps the image or presses and holds their finger on it for a brief moment. The

Figure 4: Timeline Explorer

user can zoom in and out, as well as pan the image. All metadata of the image is shown below. Favourite and hide buttons are also available for the user to use. This feature offers a more detailed view of the image, which can be useful for the user to understand the context of the image. Options for exiting the Image Viewer include a simple tap, a swipe gesture, or a back button.

4 IMPLEMENTATION

The system is implemented as a web application, which can be accessed from any device with a web browser, however, it is optimised for mobile screens. ReactJS [1] is used for the front end, while FastAPI [2] is used for the back-end for its ability to handle asynchronous requests. Both modules communicate with each other through RESTful APIs. The processing step includes segmenting the data into events, as well as extracting the metadata of the images. After that, ElasticSearch [3] is used to index the data for the retrieval process. For a complete ranking algorithm, refer to the original MyEachtra paper [26].

As for the event readers, InternLM-XComposer2 [6] is run locally on the server using 4-bit quantization to speed up the inference time. We planned to use InternLM-XComposer2 for both the text reader and the visual reader, but the model is too large and too slow to run on the server. Therefore, we decided to use OPENAI's ChatGPT [1] API for the text reader, which is faster to run and can be offloaded to the cloud.

5 EVALUATION

In this section, we look at all the lifelog questions that were used in the LSC'22 and LSC'23. A total of 17 questions were used, with a variety of information needs. The results overall are positive, with 13 out of 18 questions answered correctly, which results in an accuracy of 72.2%.

We can look at the questions that were answered correctly and incorrectly to understand the strengths and weaknesses of the system. Due to the space constraints, we only show the questions that were answered incorrectly in Table 1. Out of the five questions that were not answered correctly, two of them (ID 1 and 2) were due to the Event Retriever not being able to retrieve the relevant events. Extrapolating from the question was required to answer these questions in order to find the correct events. For example, in question 1, looking for the event of changing the office in 2020 would not return any results, but looking for other hints such as boxes or packing would return the correct event. In question 2, many events were retrieved with different office numbers, but the correct one was not retrieved. The LLMs offered slightly different answers (L2.21 and L1.22) which were other rooms that the user frequently visited. The next two questions (ID 3 and 4) posed a challenge to the system as the MLMM's inability to interpret the text in the images correctly, especially when using the global event embeddings as input. The last question (ID 5) was a visual-based question, where only two events were retrieved, but the model was not able to generate the correct answer. In this case, it is reasonable to assume the user would know the answer by looking at the images themselves.

Other questions that were answered correctly were mostly date or location-related. The model managed to provide its reasoning in 'I can't find my hand drill / electric screwdriver. Assuming that today is the 1st July 2020, when was I last using it?' This is a good indicator that the model can understand the context of the question and provide a reasonable answer. Another example is 'What date did I go homewares shopping in 2019?' where several events were retrieved, with incorrect metadata provided (which is a common issue in lifelog data). The model managed to reason that it is most likely that the user went shopping on the 24 of December, which stated 'It is common for people to go for some last-minute shopping on Christmas Eve.' It is unclear whether the model could understand the context of the question or if it was just a coincidence due to hallucination. However, repeated success in the same type of questions indicates that the model can understand the context of the question and provide a reasonable answer.

In general, the system performed well on the questions. However, due to limited interaction between the Event Retriever and the Event Reader, it is difficult to determine the exact issues. From our observation, the Event Retriever is the main bottleneck of the system. Query parsing has shown to increase the retrieval results by creating actual known-item search queries, however, it could benefit from more advanced techniques such as query expansion to suggest potential queries to the user. Post-processing has some potential to improve the system by offering sorting and grouping options, but it is not clear how much it can improve the system. Sometimes, by grouping too many events together, the system might lose all the relevant information, which is the opposite of what we want. The Event Reader, on the other hand, has shown to be more effective compared to our previous work, but there was some trade-off between speed and accuracy. Another trade-off is related to the length of the answers, which can increase the user's trust in the answer but can also overwhelm the user with too much information. Overall, the results are promising, and we are excited to see how the system can be improved in the future.

[1] https://reactjs.org/
[2] https://fastapi.tiangolo.com/
[3] https://www.elastic.co/

Table 1: Questions with wrong answers generated by the system. For each question, the type of the question and the ground truth are also shown.

ID	Question	Type	Ground Truth
1	On what date did I change my office in 2020?	Time	09/03
2	What was the number of my office door (in 2019)?	OCR	L2.42
3	Which airline did I fly with most often in 2019?	Frequency	Turkish Airlines
4	I normally wear shirts, but what is the brand of the grey t-shirt that I wore at the start of Covid-time?	OCR	Abercrombie & Fitch
5	What did I have for breakfast on Christmas Day 2019?	Object	eggs, bacon, and toast

6 CONCLUSION

In this paper, we presented MyEachtraX, a system that brings lifelog retrieval to mobile devices and assists users in finding the answers to their lifelog questions. MyEachtraX focuses on its flexibility of supporting different levels of expertise of the users by providing adjustable features. LLMs and MLLMs are used to integrate some expert-level knowledge into the system as the default settings, in order to guide the users in the right direction. A set of natural questions was used to automatically evaluate the system, and the results show that the system achieves an accuracy of 72.2%.

Future work in this area is endless. We plan to evaluate the system on a larger dataset to identify the strengths and weaknesses of this approach. User studies will be conducted to understand the user's needs and preferences and to identify areas for improvement. Better communication between the Event Retriever and the Event Reader is also necessary for the system to improve. Our query-parsing and post-processing components are still in their early stages, and we believe that they have the potential to revolutionise the way we search and retrieve lifelog data. We are excited to see how the area of lifelog retrieval will evolve in the future.

REFERENCES

[1] Josh Achiam et al. 2023. Gpt-4 technical report. *arXiv preprint*, abs/2303.08774. https://arxiv.org/abs/2303.08774 eprint: 2303.08774.

[2] Ahmed Alateeq, Mark Roantree, and Cathal Gurrin. 2023. Voxento 4.0: a more flexible visualisation and control for lifelogs. In *Proceedings of the 6th Annual ACM Lifelog Search Challenge*, 7–12.

[3] Tom B. Brown et al. 2020. Language models are few-shot learners. In *NeurIPS*.

[4] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics. DOI: 10.18653/v1/p17-1171.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*. Association for Computational Linguistics, 4171–4186.

[6] Xiaoyi Dong et al. 2024. InternLM-XComposer2: mastering free-form text-image composition and comprehension in vision-language large model, (Jan. 2024). arXiv: 2401.16420v1 [cs.CV].

[7] 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* (Lsc '18). Association for Computing Machinery, New York, NY, USA, (June 2018), 20–23. ISBN: 97-81450-357-9-6-8.

[8] Cathal Gurrin, Alan F Smeaton, Aiden R Doherty, et al. 2014. Lifelogging: personal big data. *Foundations and Trends® in information retrieval*, 8, 1, 1–125.

[9] Cathal Gurrin et al. 2022. Introduction to the fifth annual lifelog search challenge, LSC'22. In *Proc. International Conference on Multimedia Retrieval (ICMR'22)*. Newark, NJ, USA. DOI: 10.1145/3512527.3531439.

[10] Cathal Gurrin et al. 2024. Introduction to the seventh annual lifelog search challenge, LSC'24. In ACM, (June 2024). DOI: 10.1145/3652583.3658891.

[11] Nhat Hoang-Xuan, Thang-Long Nguyen-Ho, Cathal Gurrin, and Minh-Triet Tran. 2023. Lifelog discovery assistant: suggesting prompts and indexing event sequences for FIRST at LSC 2023. In *Proceedings of the 6th Annual ACM Lifelog Search Challenge* (ICMR '23). ACM, (June 2023). DOI: 10.1145/3592573.3593104.

[12] Maria Tysse Hordvik, Julie Sophie Teilstad Østby, Manoj Kesavulu, Thao-Nhu Nguyen, Tu-Khiem Le, and Duc-Tien Dang-Nguyen. 2023. LifeLens: transforming lifelog search with innovative UX/UI design. In *Proceedings of the 6th Annual ACM Lifelog Search Challenge*, 1–6.

[13] Emil Knudsen, Thomas Holstein Qvortrup, Omar Shahbaz Khan, and Björn Þór Jónsson. 2021. XQC at the lifelog search challenge 2021: interactive learning on a mobile device. In *Proceedings of the 4th Annual on Lifelog Search Challenge*, 89–93.

[14] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *NeurIPS*.

[15] Patrick Lewis et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *NeurIPS*, 33, 9459–9474.

[16] Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. 2023. Monkey: image resolution and text label are important things for large multi-modal models, (Nov. 2023). arXiv: 2311.06607v3 [cs.CV].

[17] Tien-Thanh Nguyen-Dang, Xuan-Dang Thai, Gia-Huy Vuong, Van-Son Ho, Minh-Triet Tran, Van-Tu Ninh, Minh-Khoi Pham, Tu-Khiem Le, and Graham Healy. 2023. LifeInsight: an interactive lifelog retrieval system with comprehensive spatial insights and query assistance. In *Proceedings of the 6th Annual ACM Lifelog Search Challenge* (ICMR '23). ACM, (June 2023). DOI: 10.1145/3592573.3593106.

[18] Alec Radford et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*. Pmlr, 8748–8763.

[19] Shezheng Song, Xiaopeng Li, Shasha Li, Shan Zhao, Jie Yu, Jun Ma, Xiaoguang Mao, and Weimin Zhang. 2023. How to bridge the gap between modalities: a comprehensive survey on multimodal large language model, (Nov. 2023). arXiv: 2311.07594v2 [cs.CL].

[20] Florian Spiess and Heiko Schuldt. 2022. Multimodal interactive lifelog retrieval with vitrivr-VR. In *Proceedings of the 5th Annual on Lifelog Search Challenge*, 38–42.

[21] Ly-Duyen Tran, Thanh Cong Ho, Lan Anh Pham, Binh Nguyen, Cathal Gurrin, and Liting Zhou. 2022. LLQA-Lifelog question answering dataset. In *MultiMedia Modeling: 28th International Conference, MMM 2022, Phu Quoc, Vietnam, June 6–10, 2022, Proceedings, Part I*. Springer, 217–228.

[22] 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* (LSC '20). Association for Computing Machinery, New York, NY, USA, 23–28. ISBN: 9781-450-3713-6-0.

[23] Ly-Duyen Tran, Manh-Duy Nguyen, Binh T Nguyen, and Liting Zhou. 2023. Myscéal: a deeper analysis of an interactive lifelog search engine. *Multimedia Tools and Applications*, 1–18.

[24] 2024. *Interactive question answering for Multimodal lifelog retrieval. Lecture Notes in Computer Science*. Springer Nature Switzerland, 68–81. ISBN: 9783-031-56435-2. DOI: 10.1007/978-3-031-56435-2_6.

[25] Ly-Duyen Tran et al. 2023. Comparing interactive retrieval approaches at the lifelog search challenge 2021. *IEEE Access*, 11, 30982–30995.

[26] Ly Duyen Tran, Binh Nguyen, Liting Zhou, and Cathal Gurrin. 2023. MyEachtra: event-based interactive lifelog retrieval system for LSC'23. In *Proceedings of the 6th Annual ACM Lifelog Search Challenge*, 24–29.

[27] A Waswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, A Gomez, L Kaiser, and I Polosukhin. 2017. Attention is all you need. In *Nips*.

[28] Wayne Xin Zhao et al. 2023. A survey of large language models. (2023). DOI: 10.48550/ARXIV.2303.18223.