# A RAG Approach for Multi-Modal Open-ended Lifelog Question-Answering


Quang-Linh Tran*
ADAPT Centre, School of Computing,
Dublin City University
Dublin, Ireland
linh.tran3@mail.dcu.ie

Ngo Ngoc Diep Pham
University of Science,
Vietnam National University
Ho Chi Minh City, Vietnam
21110058@student.hcmus.edu.vn

Quoc Trung Truong
University of Science,
Vietnam National University
Ho Chi Minh City, Vietnam
21110427@student.hcmus.edu.vn

Minh Hung Nguyen
University of Science,
Vietnam National University
Ho Chi Minh City, Vietnam
21110301@student.hcmus.edu.vn

Hong Cat Le
University of Science,
Vietnam National University
Ho Chi Minh City, Vietnam
21110249@student.hcmus.edu.vn

Dang Khoi Vu
University of Science,
Vietnam National University
Ho Chi Minh City, Vietnam
22280049@student.hcmus.edu.vn

Van Minh Thien Nguyen
University of Science,
Vietnam National University
Ho Chi Minh City, Vietnam
nvmthien22@clc.fitus.edu.vn

Van Kinh Nguyen
University of Science,
Vietnam National University
Ho Chi Minh City, Vietnam
22280051@student.hcmus.edu.vn

Luu Phuong Ngoc Lam Nguyen
University of Science,
Vietnam National University
Ho Chi Minh City, Vietnam
21280096@student.hcmus.edu.vn

Tan Le
University of Science,
Vietnam National University
Ho Chi Minh City, Vietnam
22110196@student.hcmus.edu.vn

Minh Phuc Dang
University of Science,
Vietnam National University
Ho Chi Minh City, Vietnam
22280064@student.hcmus.edu.vn

Binh Nguyen
University of Science,
Vietnam National University
Ho Chi Minh City, Vietnam
ngtbinh@hcmus.edu.vn

Gareth J. F. Jones
ADAPT Centre, School of Computing,
Dublin City University
Dublin, Ireland
gareth.jones@dcu.ie

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



## Abstract

Lifelogging is the passive collection, storage and analysis of daily data through wearable sensors. Question Answering (QA) for lifelog data enables natural language interactions with personal daily life records, providing insights into individual routines and behaviours. While this task has great potential for personal analytics and memory augmentation, progress has been limited due to the challenges of lifelog management, since they can comprise of enormous multi-modal data sets spanning a lifetime. We introduce a Retrieval-Augmented Generation (RAG) approach for addressing the lifelog QA task. A RAG approach first includes a retrieval model finding the correct lifelog events containing answers and then a large language model (LLM) generating answers from the questions. In addition, we construct an open-ended lifelog QA benchmark with 14,187 QA pairs to examine the RAG approach to lifelog QA. Using an embedding-based retrieval approach, our lifelog context retriever achieves a performance of 77.67% Recall@5 and 94.35% Recall@20 using an embedding-based retrieval approach with the Stella 1.5B model. Combined with the Mistral 7B model, the model achieves scores of 39.54% ROUGE-L and 3.475 Accuracy on a scale of 5 scored by GPT-4o. This approach potentially provides an effective approach to lifelog QA with high performance that does not require fine-tuning.



*Corresponding author.





*ICMR '25, Chicago, IL, USA*





ACM ISBN 979-8-4007-1877-9/2025/06
https://doi.org/10.1145/3731715.3733263



## CCS Concepts

• **Computing methodologies** → **Natural language generation**.

## Keywords

Lifelog Question Answering; Multi-modal Question Answering Dataset; Large Language Models; Retrieval-Augmented Generation

**ACM Reference Format:**
Quang-Linh Tran, Ngo Ngoc Diep Pham, Quoc Trung Truong, Minh Hung Nguyen, Hong Cat Le, Dang Khoi Vu, Van Minh Thien Nguyen, Van Kinh Nguyen, Luu Phuong Ngoc Lam Nguyen, Tan Le, Minh Phuc Dang, Binh Nguyen, Gareth J. F. Jones, and Cathal Gurrin. 2025. A RAG Approach for Multi-Modal Open-ended Lifelog Question-Answering. In *Proceedings of the 2025 International Conference on Multimedia Retrieval (ICMR '25), June 30-July 3, 2025, Chicago, IL, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3731715.3733263

## 1 Introduction

Lifelogging refers to the process of passively collecting, storing, and analysing an individual's daily life data through various wearable cameras, sensors, and mobile devices [7]. Lifelog data includes diverse data ranging from Point-of-View (PoV) images, biometrics, to GPS data. The challenge of lifelog remains in the volume of lifelog data spanning from several years to an entire life, the variety of data types with multi-modality, and the velocity of data increasing daily. Lifelogging also brings a lot of benefits in health monitoring [14] and memory augmentation [13]. Lifelog retrieval is the task of finding events in lifelog collections, and it is an active research area with several ongoing benchmark workshops such as the Lifelog Search Challenge (LSC) [8], and NTCIR Lifelog Task [43]. These workshops gather many lifelog retrieval systems [24, 32, 34] performing lifelog event searching on a benchmark dataset and evaluating to find state-of-the-art (SOTA) retrieval systems.

Question Answering (QA) on lifelog data is a task of asking and answering questions on lifelog data through the use of natural language queries. LifelogQA enhances the retrieval task by finding the events and providing answers to the question. In addition, lifelog QA offers personal insights into lifestyle and memorable events through an ask-and-answer manner. Although the applications of Lifelog QA are enormous, there are still two challenges in the QA task for lifelog data. Firstly, due to the large volume of lifelog data, it is necessary to find relevant information for the question before deriving answers. The accuracy of retrieval models significantly affects the correctness of the answer. Secondly, the temporal and multi-modal lifelog data challenge the QA model's reasoning for complicated questions. For example, "Who did I talk with at the ICMR conference last year?". The QA model first finds the events of having a conversation at last year's ICMR conference and then analyses the events before providing an answer. The QA task in lifelog has been introduced in the LSC challenge since 2022 [9] and remains a challenging task with no specific QA models specialised for it.

Retrieval-augmented generation (RAG) [18] combines information retrieval with text generation, which helps to incorporate external knowledge into language models through prompts to accommodate the drawback of static knowledge of language models. RAG proves its effectiveness and efficiency in various domains that need external knowledge, such as Medical QA [38] and Financial QA [1]. Lifelog data is external knowledge to LLM because this data is not available in LLM's training data. It is challenging for LLM to understand a person's life and generate personalised answers to questions, so using RAG to retrieve relevant information and incorporate it into prompts for LLM helps to address the problem of lacking knowledge. We propose a RAG system for the lifelog QA task based on several retrieval approaches and LLM. For the Lifelog QA task, RAG retrieves relevant lifelog context (an event of life, such as eating or walking, and metadata, such as time and location) to the question from an extensive lifelog data collection and generates an answer to the question. An example can be seen in Figure 1. This approach can help to address the challenge of vast lifelog data and the lack of knowledge in LLM to generate answers.

This paper proposes a RAG approach to address Lifelog QA and introduces a novel dataset, which is open-ended and practical based on real lifelog data, to examine the effectiveness of the RAG approach. There are three main contributions in this paper:

(1) To bridge the gap in existing datasets for lifelog QA, we construct the OpenLifelogQA with 14,187 QA in various difficulty levels.
(2) We propose a RAG approach for the lifelog QA task with a detailed retrieval and generation pipeline.
(3) We carry out extensive experiments and provide a comprehensive analysis of different aspects of RAG for lifelog QA.

## 2 Related Works

Although lifelog retrieval has been an active area of research in recent years with several retrieval systems [24, 32], there are only a few studies on automatic and interactive Lifelog QA. MyEachtraX [30] is a Lifelog QA app for mobile devices that participated in the Lifelog Search Challenge 2024 [8] and achieved a competitive performance against other systems on the interactive benchmark. Tran et al. [31] proposed an interactive lifelog QA system by employing an open-domain QA pipeline [3] with a retriever-reader architecture to create an interactive lifelog QA system. There are only three available datasets for the automatic benchmarking in this task. Firstly, Tran et al. [29] constructed an LLQA dataset, an augmented 85-day lifelog collection, and over 15,000 multiple-choice questions. The pilot experiment for baseline models such as TVQA [15], and Sequence to Sequence (S2S) [2]. The accuracy for TVQA is 63.38% accuracy on Yes/No questions and 61.36% accuracy on multiple-choice questions. The performance of the S2S model is lower, at 50.66% for Yes/No questions and 36.26% for multiple-choice questions. Secondly, MemoriQA [33] is a small-scaled open-ended lifelog dataset with only 61 days of lifelog and 3,644 QA pairs. There are only 1,925 events, and they are not open-sourced for use. Another notable lifelog QA dataset from Meta Inc. and Cornell University is TimelineQA [28]. They constructed an automatic lifelog data generation model to create synthetic lifelog episodes of imaginary people, from milestones such as marriage and graduation to daily activities like talking to friends or having dinner. A baseline experiment is carried out for three approaches: RAG [18], ExtractiveQA [12] for atomic questions and TableQA with Tapex [21] and BART [16] variants for multi-hop and aggregation questions. Experimental results show that an extractive QA model significantly outperforms a RAG QA model with 94.8% F1 compared to 84.4% F1 for atomic queries. Multi-hop queries involving aggregates show that the best result is obtained with a TableQA technique with a Tapex-large model at only 59.0% F1. Although RAG was used in this study, it performed poorly due to the extractive nature of the benchmark. This research enhances this approach by using a specialised indexing, retrieval and generation approach for lifelog data.

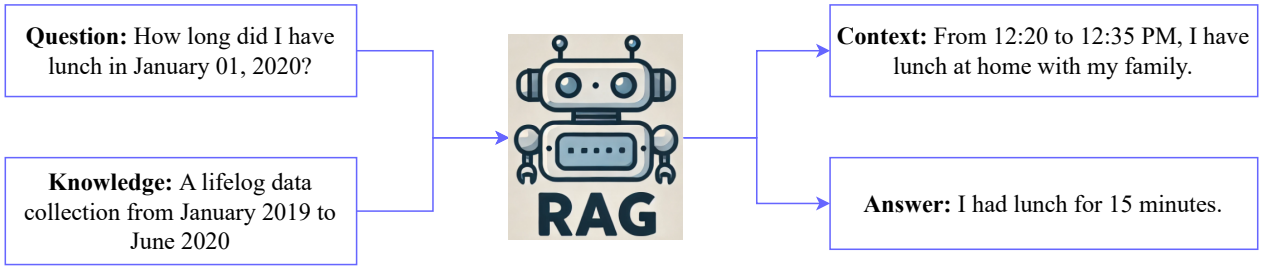



**Figure 1: A RAG approach example for Open-ended Lifelog QA. RAG receives a question as input and searches over a lifelog data collection to retrieve the context for the question and provide a textual answer.**

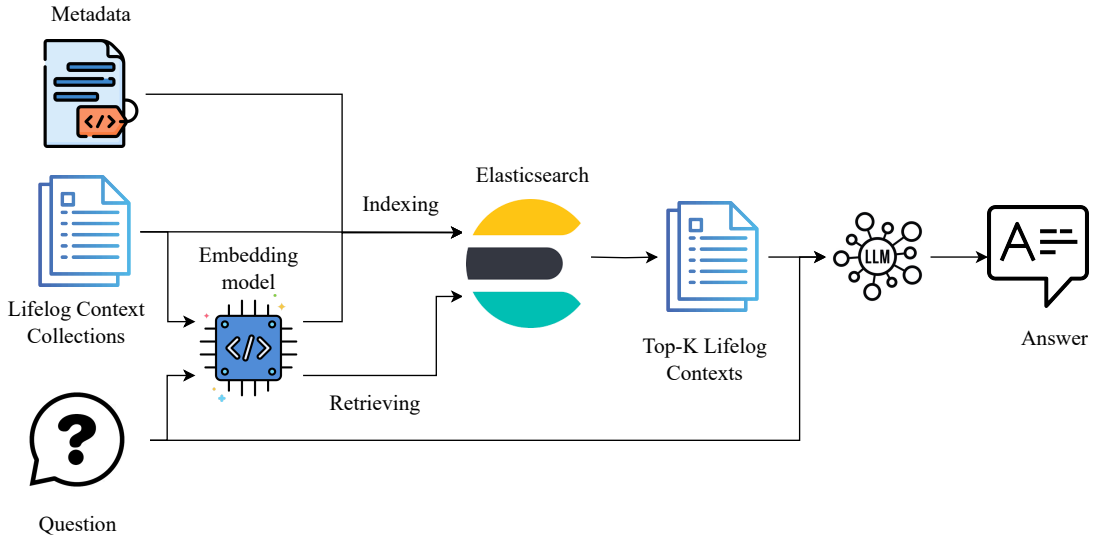



**Figure 2: RAG Approach for Lifelog QA**

RAG [18] has emerged as a powerful approach for knowledge-intensive tasks in natural language processing (NLP) by integrating retrieval-based knowledge into LLM generation. This integration addresses challenges such as hallucinations and outdated knowledge in LLM. The RAG framework has been refining the architecture to further enhance both retrieval and generation accuracy. For instance, InstructRAG [37] proposes explicit denoising through self-synthesised rationales to enhance generation accuracy by instructing models to explain how answers are derived from retrieved documents. DuetRAG [11] introduces a collaborative framework that integrates domain fine-tuning with RAG models to improve knowledge retrieval quality, particularly in complex domains like HotPot QA [40]. Several studies have also explored RAG's potential in various domains, such as open-domain question answering [27], medical QA [39], and financial QA [41]. In lifelogging, RAG-based models have been explored to generate answers for questions based on personal data [28], but they underperform the extractive approach because of the extractive nature of answers in the TimelineQA dataset. To further examine the capability of RAG in lifelog QA, we construct an open-ended lifelog QA dataset with generative answers.

## 3 Methodology

In this section, we provide details about the RAG approach for the Lifelog QA task. Firstly, we describe the problem formulation to clarify the lifelog task requirements. Secondly, the RAG approach for Lifelog QA is discussed with a detailed architecture. Subsequently, the retrieval and generation model is introduced to illustrate the components of RAG.

### 3.1 Problem formulation

Given the requirement of Lifelog QA task to generate answers for questions, we have a set of N questions $Q = \{q_1, q_2, ..., q_n\}$ and a set of corresponding answers $A = \{a_1, a_2, ..., a_n\}$ in which $a_1$ is the answer for question $q_1$. We also have a lifelog collection segmented

into events (a sequence of images illustrating a single activity, such as eating or walking). Each event description is added to the time and location in the metadata to form a lifelog context, such as "From 12:20 to 12:35 PM 01/01/2020, I had lunch at home with my family". We have a set of M contexts $C = \{c_1, c_2, ..., c_m\}$.

For a given question q, we first retrieve all contexts relevant to the question $C_q$. The relevance is measured using metrics based on the retrieval model, such as BM25 or cosine similarity.

$$C_q = Retrieval(q)$$

From the set of contexts $C_q$ and the question $q$, we formulate a prompt and provide it to an LLM to generate the answer $a$.

$$a = LLM(prompt(C_q, q))$$

From the problem formulation, we have two different tasks in the RAG approach, including context retrieval and answer generation. The goal of context retrieval is to find accurate contexts needed for questions, while the target of answer generation is to generate the correct answer for the question.

## 3.2 RAG

Figure 2 presents RAG's indexing, retrieving and question-answering process. In the indexing stage, lifelog contexts are indexed to Elasticsearch[1], a robust vector database. Each lifelog context is an entity with metadata such as time and location, event description, keyframe image, and embedding vector of an embedding model on the event description. In the retrieval stage, a question is processed into a query by removing the question word and mark. A ranked list of lifelog contexts is retrieved for the query and is ready for the next stage. Details on how a ranked list is retrieved are described in section 3.3. In the answering stage, the question and ranked list of lifelog contexts are formulated into a prompt. This prompt is provided to the LLM to generate the answer to the question. Details about LLM and the generation stage are described in section 3.4.

## 3.3 Retrieval models

Retrieval models play an important role in RAG systems, which retrieve information from the lifelog context collection to find relevant contexts for questions. The model first transforms a question into a query by removing question words and punctuation marks. The retrieval model then retrieves the most relevant lifelog contexts by using BM25 or vector-based approaches.

**BM25** [26] is a classic lexical-based algorithm that ranks documents based on term frequency and inverse document frequency, weighing how often query terms appear in a document against their rarity throughout the collection. Although BM25 is good at exact keyword matching and is computationally efficient, it falls short when dealing with synonyms or conceptually related terms. This approach is a baseline for comparison with another vector-based approach.

**Stella-en-1.5B-v5**[2] is an embedding model that transforms the textual query into embedding vector. This vector, along with the embedding vectors of lifelog contexts indexed in Elasticsearch, is used to compute the cosine similarity. The ranked list is retrieved by sorting the cosine similarity in descending order. This approach captures deeper semantic relationships and better handles vocabulary mismatch, allowing it to find relevant documents even when they use different but related terms. We choose this embedding model because it is a general embedding model that ranks No. 5 on the Massive Text Embedding Benchmark (MTEB benchmark)(as of November 30, 2024) [22] with a score of 71.19 across all 56 text retrieval tasks.

For each question, a retrieval model finds relevant information and creates a ranked list of lifelog contexts for the answer generation task.

## 3.4 Large language models

LLM is a transformer-based neural network model designed to generate text [10, 23]. LLM is trained on vast and diverse datasets, including websites [3], books, news articles, and code, and consists of billions of parameters. With a wide range of applications, LLM is expected to effectively solve the questions in lifelog when enough contexts are provided. In the RAG approach for lifelog QA, LLM receives a prompt of instruction, retrieved contexts and a question as input and generates an answer. We use two LLMs for the experiment, including Mistral 7B Instruct V0.3 [10] and Llama 3.1 8B Instruct [6].

**Mistral 7B Instruct V0.3** is a language model introduced by the Mistral.AI Research team. It has 7 billion parameters and employs advanced techniques like a mixture of experts (MoE) to achieve high performance with reduced computational overhead. The model is fine-tuned for instruction-following tasks to deliver precise, context-aware responses. Its performance in various benchmarks is superior to Llama2 7B and Llama2 13B. In addition, this model also proves reasoning capability in coding problems. Hence, we choose the Mistral 7B model to do the generation part in RAG to solve lifelog QA questions, especially aggregation questions that need reasoning.

**Llama3.1 8B Instruct** developed by Meta is a state-of-the-art instruction-tuned language model optimized for multilingual conversational use. This model is trained on a massive dataset of over 15 trillion tokens, including both publicly available content and synthetic instruction examples. This model was released in July 2024 and achieved strong performance on benchmarks such as MMLU and HumanEval. We use this model to compare Mistral 7B Instruct V0.2 to comprehensively view LLM in lifelog QA.

These two models are provided a prompt with lifelog contexts and a question. We experiment with different types of instruction in the prompt, including zero-shot (only instructions for the task), few-shot (instruction and several examples of QA), and Chain-of-thought (CoT) [36] (instruction and several examples of intermediate explanation towards the answer).

# 4 OpenLifelogQA Dataset

To evaluate the performance of the RAG approach in lifelog QA, we construct an open-ended lifelog QA dataset. This dataset is distinctive from previous datasets with generative answers, diverse difficulty levels and realistic lifelog. We describe the OpenLifelogQA dataset, from the source of lifelog data to annotation. In addition, we present key information about the dataset.

[1] https://www.elastic.co/

[2] https://huggingface.co/dunzhang/stella_en_1.5B_v5

[3] https://www.wikipedia.org/

**Table 1: Statistics on three sets of OpenLifelogQA.**

| Set | # atomic question | # temporal question | # aggregation question | # context events | Avg question length | Avg answer length |
|---|---|---|---|---|---|---|
| Training | 6578 | 946 | 3653 | 1.64 | 12.74 | 6.67 |
| Validation | 895 | 131 | 476 | 1.75 | 11.99 | 5.74 |
| Testing | 882 | 156 | 470 | 1.75 | 12.23 | 5.73 |

**Table 2: Lifelog QA datasets comparison.**

| Dataset | # QA | Lifelog data | Question format | # Event description | Image | Time & Location | Open-sourced |
|---|---|---|---|---|---|---|---|
| LLQA [29] | 15,065 | 85 days | Multiple-choice | None | ✓ | ✓ | ✓ |
| Timeline QA [28] | 600,000 | Synthetic | Open-ended | 128,023,476 | | ✓ | ✓ |
| MemoriQA [33] | 3,644 | 61 days | Open-ended | 1,925 | ✓ | ✓ | |
| OpenLifelogQA | 14,187 | 514 days | Open-ended | 27,705 | ✓ | ✓ | ✓ |

## 4.1 Dataset construction

The lifelog dataset used for QA annotation is from the ACM LSC'24 [8]. This dataset contains 725K real-world PoV lifelog images from an anonymous lifelogger and associated metadata such as location and time, ranging over 18 months from January 2019 to June 2020. Although annotating a QA dataset is labour-intensive, human annotators for QA pairs in lifelog help to create high-quality and realistic questions. For this reason, we recruited 10 volunteers, all undergraduate students majoring in Data Science and Information Technology. The volunteers received training through two sessions covering lifelog concepts, lifelog QA and the annotation process. A comprehensive annotation guideline was also provided, detailing the overall process and addressing common issues such as discrepancies between local time and GMT and location confusion.

After intensive training for annotators, each annotator is assigned 55 dates of lifelog data to annotate. This means each annotator is tasked with three dates of lifelog to annotate and one date to evaluate the annotation quality for each month of lifelog data. There are four stages in the annotation process. In the first stage, annotators review all lifelog images and metadata for the date. In the second stage, annotators generate the description for each event, a sequence of images illustrating the lifelogger in a single activity. In the next stage, pairs of QA are created by the annotator and the GPT-4o model [23], a SOTA LLM provided by OpenAI. Specifically, the annotators' task is to generate 10 QA pairs and provide the events in stage 2 that contain information for the answer. For each QA pair, one or more events are linked to the QA. For the process of GPT-4o generating QA, this model is asked to generate 20 pairs of QA and return a JSON object of triples: question-answer-eventid that contains information for the answer. Notably, annotators and the GPT-4o model have freely generated questions without restrictions on question words. However, to increase the difficulty level of questions, we ask annotators and the GPT-4o model to annotate aggregation questions as many as possible. After this stage, 30 pairs of QA are generated each day in the lifelog dataset. In the final stage, annotators check the accuracy and quality of GPT-4o-generated QA and edit any errors. This stage aims to improve the quality of QA pairs and avoid errors from LLM-generated QA.

## 4.2 Dataset information

After the annotation process, we have a lifelog QA dataset consisting of 27,705 event descriptions and 14,187 QA pairs. With a total of 514 days of lifelog data, there are about 54 events and 28 questions per day. The number of events and QA pairs is independent because annotators generate QA from a pooled event. The dataset is divided into three subsets: training, validation, and testing, with data from April and May 2020 for validation, March and June 2020 for testing, and the rest for training. This splitting strategy aims to avoid data leakage in training. The training set contains 11,159 QA pairs, while the validation and testing sets each include approximately 1,500 QA pairs. Table 1 provides detailed statistics for these subsets.

The questions are categorised into three types: *atomic*, *temporal*, and *aggregation*, based on classifications from previous research [28, 29]. *Atomic questions* require only a single context to answer, for example, "What time did I have lunch yesterday?". This type makes up the majority of the dataset, with 8,355 questions. *Temporal questions* relate to the temporal nature of lifelog data, asking about preceding or subsequent events relative to a known event, for example, "What did I do after having lunch yesterday?". There are 1,233 questions in this type. *Aggregation questions* need aggregating information from multiple events to calculate answers, such as determining durations, counts, or totals, and this category includes 4,599 questions. An example of an aggregation question is: "How long did I have lunch yesterday?".

This dataset provides diverse lifelog QA pairs, reflecting different question types and complexities. The dominance of atomic questions indicates the prevalence of straightforward, single-event queries in lifelog data. However, temporal and aggregation questions emphasise the importance of multi-event reasoning and temporal analysis in lifelog QA. The distribution of question words reveals that the dataset aligns with practical lifelog use cases, focusing on core questions about "what," "where," and "when," while less emphasis is placed on exploratory or less typical queries.

## 4.3 Comparison with existing datasets

As the motivation for constructing this dataset is to focus on generative and open-ended QA, we compare this dataset to previous

benchmarks to illustrate our novel work. Table 2 shows information about four lifelog QA datasets. Our OpenLifelogQA dataset is a more comprehensive resource for lifelog QA research than previous works. Except for the Timeline QA dataset, which is created automatically from templates, our OpenLifelogQA dataset has more event descriptions and lifelog data than other datasets. In addition, OpenLifelogQA is constructed with open-ended questions and contains images, as well as time and location information. Notably, the dataset is open-sourced for the event descriptions and QA pairs for further exploration by the research community.

## 5 Experiment

This section presents the experimental setup and the results of the baseline RAG approach. Additionally, we analyse the impact of various retrieval settings and prompt types. The section also includes a discussion on the effectiveness of the RAG approach.

### 5.1 Settings

From the training, validation, and testing sets of the OpenLifelogQA dataset, we use the validation set to set up the hyper-parameters of retrieval and generation before predicting the test set. As each question in the dataset has a specific date, for example, "How long did I have lunch on January 01, 2020?", there is a filter for retrieval on a date. For the retrieval setting, we set the top K retrieved lifelog contexts, with K from 1 to 100, as the maximum context per day is around 100. The size of the vector embedding of queries and lifelog context is 1024. We use approximate nearest neighbour search in Elasticsearch with a number of neighbours of 1000 for vector-based search. We set up experiments on different retrieved contexts, from the top 20, the top 50, the all-day context (all contexts on the date of the questions), and the oracle (ground-truth context of QA). For the LLM model, we set the temperature to 0.5, top P to 0.5, and maximum generated tokens to 256. Details about setting up RAG for lifelog QA and reproducing experiments are published in GitHub [4]. All the experiments are conducted on an NVIDIA A100 GPU with 80 GB.

We use the BART [17] and the T5 [25] models as baseline language models to compare with LLM. BART is a denoising autoencoder that combines bidirectional and autoregressive transformers for pretraining, making it effective for various generative tasks. T5, on the other hand, frames all NLP tasks as a text-to-text problem and is pretrained on a large corpus using a span corruption objective. Both models represent strong pretrained sequence-to-sequence architectures, yet they are considered relatively small compared to modern LLMs like GPT-4 or LLama. We aim to highlight the challenge of the lifelog QA task, which requires a generative nature to address this problem.

To evaluate the performance of retrieval models, we use the Recall@K (R@K) and Precision@K (P@K) metrics to compare the retrieved ranked list and ground-truth contexts (generated in stage 3 of annotations). This metric helps to evaluate the correctness of retrieved contexts. Three approaches are commonly used to evaluate lifelog QA models' accuracy for the answer generation task: lexical matching, semantic matching, and LLM-based evaluation. Lexical matching measures word overlap between predictions and ground truth using metrics like Exact Match, F1 score, and ROUGE [20]. ROUGE evaluates n-gram overlap and sequence similarity. We use this metric for lexical matching between predicted and ground truth answers. While compelling, it can miss cases of semantic equivalence with lexical variation. Semantic matching addresses this limitation by evaluating the meaning rather than the wording of the text. BERTScore [42] is a popular metric, leveraging BERT embeddings [5] to calculate semantic similarity between ground-truth and predicted answers. LLM-based evaluation [4] uses LLM like GPT-4o [23] to rate the accuracy of predictions on a 1-to-5 scale. ROUGE, BERTScore, and LLM-based evaluation are chosen to evaluate the performance of lifelog QA models.

[4]https://github.com/linh222/rag_openlifelog_qa

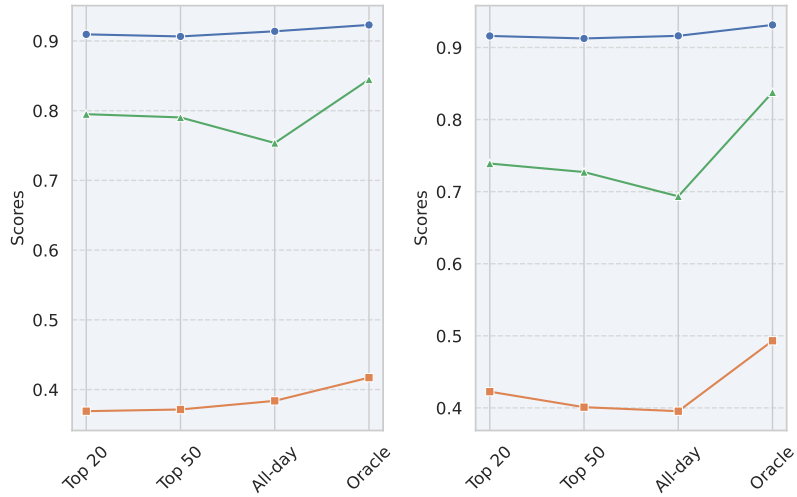
### 5.2 Results

Table 3 provides the retrieval performance of two approaches for the context retrieval task. BM25 performs worse than the Stella-en-1.5B-v5 model in all metrics. BM25 only achieves 22.99% Recall@1 and 91.38% Recall@20, compared to 37.65% Recall@1 and 94.35% Recall@20 in the Stella-en-1.5B-v5 model. The result indicates that the lexical matching approach finds contexts for lifelog QA less accurate. It is because of the mismatched format of temporal information in lifelog contexts and queries. The diversity of date and time, such as 9:24 PM vs 21:24, January 01, 2020, vs. 01/01/2020, makes it challenging for BM25 to find the correct contexts. Meanwhile, as the Stella-en-1.5B-v5 model is trained on massive data with various date and time formats, it can retrieve this information better. Another reason for the low performance is the lack of information in the question. For example, questions like "What did I do at 3 PM on January 01, 2020?" do not contain any information about specific activity but only the time at 3 PM. If the retrieval model fails to understand the time, it fails to find the correct context.

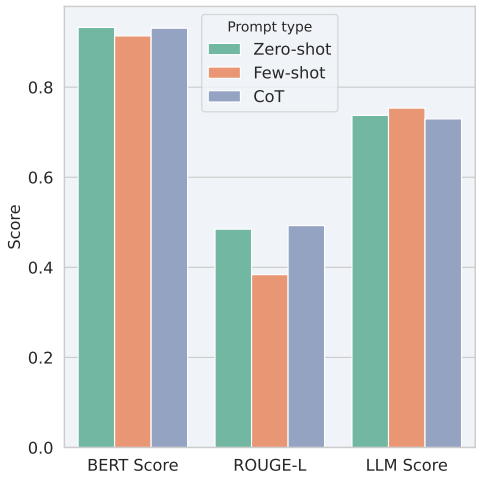
**Table 3: Context retrieval results.**

| Metrics | BM25 | Stella-en-1.5B-v5 |
|---|---|---|
| R@1 | 0.2299 | 0.3765 |
| R@5 | 0.6829 | 0.7767 |
| R@10 | 0.8171 | 0.8801 |
| R@20 | 0.9138 | 0.9435 |
| R@50 | 0.9859 | 0.9918 |
| P@1 | 0.309 | 0.5265 |
| P@5 | 0.1999 | 0.2479 |
| P@10 | 0.1273 | 0.1476 |
| P@20 | 0.0795 | 0.0854 |
| P@50 | 0.0518 | 0.0524 |

Table 4 represents the performance of two LLMs and two baseline models in answering questions from all-day contexts. We can clearly see that the two baseline models perform badly in this task. T5 is better than BART with 83.56% BERT Score and 2.4682 LLM Score. However, both models achieved less than 10% ROUGE-L Score, indicating low performance in generating answers following the format of ground-truth answers. On the other hand, the Mistral 7B model performs better than the Llama 3.1 8B model in BERT Score and ROUGE-L metrics, but Llama 3.1 8B has a high LLM Score. Mistral 7B achieves a 91.61% BERT Score, 39.54% ROUGE-L Score,




**(a) Llama 3.1 (left) and Mistral (right) performance on different context settings. Circled blue line is the BERT Score, triangular green line is the scaled LLM score, and squared orange line is ROUGE-L score.**




**(b) Performance of the Llama 3.1 model in different prompts (Scaled LLM Score). Few-shot and CoT prompts have five examples of QA and Q-intermediate thought-A, respectively.**

**Figure 3: Analysis on context settings and different prompt types to LLM models**

and 3.475 LLM Score. This score indicates that the Mistral 7B model follows the instructions to generate the aligned format with ground-truth answers, but the accuracy is not as high as that of Llama 3.1. In addition, although the BERT Score is high, the ROUGE-L and LLM Score are about average. It indicates that the semantics of predicted and ground-truth answers are highly correlated, but the words and accuracy of predicted answers are not high.

**Table 4: Experimental result of two LLM and baseline models for all-day contexts (all lifelog contexts on the day of questions)**

| Models | BERT Score | ROUGE-L | LLM Score |
|---|---|---|---|
| BART | 0.8131 | 0.0529 | 2.0385 |
| T5 | 0.8356 | 0.0977 | 2.4682 |
| Mistral-7B | 0.9161 | 0.3954 | 3.475 |
| Llama-3.1-8B | 0.9138 | 0.3838 | 3.7679 |

## 5.3 Analysis

*5.3.1 Performance on different question types.* To examine the performance of LLM models in different types of questions, we aggregate the metrics by question types. Table 5 shows the performance of the Llama-3.1 model in all-day context settings for different types of questions. This model achieves the highest score in temporal questions with 43.44% ROUGE-L and 4.1282 LLM Score, followed by Atomic with 36.16% ROUGE-L and 3.7449 LLM Score. Aggregation questions are the most challenging type of questions, and Llama-3.1 achieves the worst performance on this type of question with only a 3.6915 LLM Score. Although the ROUGE-L of aggregation questions is higher than the figure for atomic questions, it only indicates the lexical matching of predicted and ground-truth answers, not the accuracy of meaning. This result is expected because the aggregation question is difficult due to the complex reasoning required to calculate the performance from a lifelong context. There remains room for improvement in LLM reasoning in lifelog QA aggregation questions.

**Table 5: Experimental results for different types of questions in the Llama-3.1 model for all-day contexts.**

| Question types | BERT Score | ROUGE-L | LLM Score |
|---|---|---|---|
| Atomic | 0.9140 | 0.3616 | 3.7449 |
| Temporal | 0.9293 | 0.4344 | 4.1282 |
| Aggregation | 0.9085 | 0.4086 | 3.6915 |

*5.3.2 Performance in different context settings.* To examine the influence of contexts on answer generation, we analyse the performance of LLM in different context settings, from the top 20, top 50, and all-day contexts to oracle contexts. Figure 3a illustrates the performance of LLM models in lifelog QA tasks when different contexts are provided in the prompt. It can be seen that when the number of contexts increases, the performance of all metrics increases, except for the all-day context. It is expected because more context in the prompt provides more information for the LLM to process and generate the answer. However, when we provide oracle contexts, which are the ground-truth contexts generated in annotations, LLM models cannot always correctly generate answers. It indicates that LLM's challenge is not understanding the context enough to generate correct answers.

*5.3.3 Performance on different prompts.* As discussed in the section 3.2 with three different types of prompts, including zero-shot learning, few-shot learning and Cot, we examine the influence of

**Table 6: Some examples of LLama 3.1 predictions**

| Question | Oracle Contexts | Ground-truth Answer | Predicted Answer | LLM Score |
|---|---|---|---|---|
| How long did I prepare dinner on March 15, 2020? | From 05:01 PM to 05:54 PM on March 15, 2020, I finish checking and prepare dinner at Home. | I prepared dinner for about 52 minutes. | I prepared dinner in 53 minutes. | 4 |
| Where did I go at 7:08 AM on June 4, 2020? | From 07:08 AM to 07:09 AM on June 04, 2020, I went to the gate to receive a courier package at Home. | I went to the gate to receive a courier package. | Home | 1 |

prompts on answer generation by experimenting with different prompts. Figure 3b illustrates the performance of the Llama 3.1 model in different types of prompts. Notably, the prompt does not significantly affect the performance of the models. Few-shot learning and CoT are emerging capabilities of LLM, seen on hundreds of billions of parameters [35] so that the 7B and 8B models cannot follow examples in the prompt. CoT prompt provides a slightly better performance than the other two prompts. However, few-shot learning is expected to increase the performance of LLM models, but it actually decreases. This may indicate that the size of the LLM models is not big enough to allow them to follow the examples in the instructions. Further experiments are necessary to evaluate the effectiveness of prompts in lifelog QA in bigger LLM models.

*5.3.4 Qualitative examples.* Table 6 shows some examples of correct and wrong predictions from Llama 3.1 models on oracle contexts. In the first example, the model correctly inferred the duration of the event with only a slight discrepancy in phrasing, earning a high LLM score. However, in the second example, the model failed to capture the full semantic meaning of the event, resulting in an incomplete answer and a much lower score. It misunderstands the intention of where to go to provide a correct answer. These examples highlight the weakness and robustness of LLM in the Lifelog QA task and show the room for improvement.

## 5.4 Discussion

*5.4.1 Scalability of RAG approach for Lifelog QA.* To address the challenge of the velocity of lifelog data, which is growing every day, the RAG approach is highly scalable to address this problem. As new lifelog data is generated, automatic event segmentation and description creation can be done using advanced image-to-text models, such as BLIP2 [19]. These descriptions can be processed and embedded into vector embeddings before indexing them in databases such as Elasticsearch. Furthermore, this approach supports incremental updates, for which only new data needs to be indexed. The RAG approach allows retrieval models and LLMs to handle new data without additional fine-tuning.

*5.4.2 Ethical and privacy considerations.* The deployment of lifelog QA systems introduces significant ethical and privacy challenges due to the sensitive nature of personal data involved. Lifelogs often include intimate details of an individual's daily life, such as health information, social interactions, and location history. Ensuring the security and confidentiality of this data requires robust mechanisms for encryption, access control, and secure storage. For data usage in this dataset, we can provide event descriptions and QA when fulfilling usage agreements. The lifelog images are from the LSC challenge, so please contact them for the images. Furthermore, since LLM is not trained on personalised data, its responses may reflect generic biases or incorrect assumptions, particularly when questions lack sufficient context.

*5.4.3 Limitations.* There is a limitation in the performance of different prompt types. The evaluation indicates differences in the effectiveness of zero-shot, few-shot, and CoT prompts. This evaluation suggests that the LLM model with small-sized parameters may lack the capacity to exploit these prompt engineering techniques. Larger models or fine-tuned variants could improve performance, particularly for complex reasoning tasks like aggregation and temporal questions. Additionally, using GPT-4o and human annotators for dataset creation introduces variability in dataset quality. While annotators undergo training, inconsistencies in event descriptions or QA pairs may happen due to subjective interpretations. With GPT-4o and manual edits, the semi-automated annotation process could result in errors or biases from the language model. Future efforts should focus on refining annotation guidelines and employing more quality control measures to enhance dataset reliability.

## 6 Conclusion

This study introduced a RAG approach for lifelog QA and an OpenLifelogQA dataset designed to address the unique challenges of question-answering on lifelog data. OpenLifelogQA is a comprehensive resource for improving QA models in lifelogs, with 14,187 diverse question-answer pairs collected over 18 months of realistic lifelogs. Our implementation of a RAG approach showed the potential of embedding-based retrieval in achieving robust performance, with high retrieval accuracy (77.67% R@5 and 99.18% R@50) and competitive QA accuracy (39.54% ROUGE-L and 3.475 LLM Score). These results prove the feasibility of retrieval and generative models for lifelog QA tasks. Future work will focus on exploration into advanced multi-modal retrieval techniques and fine-tuning generative models to handle complex Lifelog QA.

## Acknowledgments

This research was conducted with the financial support of Research Ireland at ADAPT, the Research Ireland Centre for AI-Driven Digital Content Technology at Dublin City University [13/RC/21-06_P2].

## References


[1] Muhammad Arslan, Hussam Ghanem, Saba Munawar, and Christophe Cruz. 2024. A Survey on RAG with LLMs. *Procedia Computer Science* 246 (2024), 3781–3790. doi:10.1016/j.procs.2024.09.178 28th International Conference on Knowledge Based and Intelligent information and Engineering Systems (KES 2024).

[2] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2020. UniLMv2: Pseudo-Masked Language Models for Unified Language Model Pre-Training. arXiv:2002.12804 [cs.CL] https://arxiv.org/abs/2002.12804

[3] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to Answer Open-Domain Questions. arXiv:1704.00051 [cs.CL] https://arxiv.org/abs/1704.00051

[4] Michael Desmond, Zahra Ashktorab, Qian Pan, Casey Dugan, and James M. Johnson. 2024. EvaluLLM: LLM assisted evaluation of generative outputs. In *Companion Proceedings of the 29th International Conference on Intelligent User Interfaces* (Greenville, SC, USA) *(IUI '24 Companion)*. Association for Computing Machinery, New York, NY, USA, 30–32. doi:10.1145/3640544.3645216

[5] Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[6] Abhimanyu Dubey et al. 2024. The Llama 3 Herd of Models. arXiv:2407.21783 [cs.AI] https://arxiv.org/abs/2407.21783

[7] Cathal Gurrin, Alan F. Smeaton, and Aiden R. Doherty. 2014. . doi:10.1561/1500000033

[8] Cathal Gurrin, Liting Zhou, Graham Healy, Werner Bailer, Duc-Tien Dang Nguyen, Steve Hodges, Björn Þór Jónsson, Jakub Lokoč, Luca Rossetto, Minh-Triet Tran, and Klaus Schöffmann. 2024. Introduction to the Seventh Annual Lifelog Search Challenge, LSC'24. In *Proceedings of the 2024 International Conference on Multimedia Retrieval* (Phuket, Thailand) *(ICMR '24)*. Association for Computing Machinery, New York, NY, USA, 1334–1335. doi:10.1145/3652583.3658891

[9] Cathal Gurrin, Liting Zhou, Graham Healy, Björn Þór Jónsson, Duc-Tien Dang-Nguyen, Jakub Lokoć, Minh-Triet Tran, Wolfgang Hürst, Luca Rossetto, and Klaus Schöffmann. 2022. Introduction to the Fifth Annual Lifelog Search Challenge, LSC'22. In *Proceedings of the 2022 International Conference on Multimedia Retrieval* (Newark, NJ, USA) *(ICMR '22)*. Association for Computing Machinery, New York, NY, USA, 685–687. doi:10.1145/3512527.3531439

[10] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7B. arXiv:2310.06825 [cs.CL] https://arxiv.org/abs/2310.06825

[11] Dian Jiao, Li Cai, Jingsheng Huang, Wenqiao Zhang, Siliang Tang, and Yueting Zhuang. 2024. DuetRAG: Collaborative Retrieval-Augmented Generation. arXiv:2405.13002 [cs.CL] https://arxiv.org/abs/2405.13002

[12] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, Online, 6769–6781. doi:10.18653/v1/2020.emnlp-main.550

[13] Basel Kikhia, Josef Hallberg, Johan E Bengtsson, Stefan Savenstedt, and Kare Synnes. 2010. Building digital life stories for memory support. *International journal of Computers in Healthcare* 1, 2 (2010), 161–176.

[14] Seongjung Kim, Seongkyu Yeom, Oh-Jin Kwon, Dongil Shin, and Dongkyoo Shin. 2018. Ubiquitous Healthcare System for Analysis of Chronic Patients' Biological and Lifelog Data. *IEEE Access* 6 (2018), 8909–8915. doi:10.1109/ACCESS.2018.2805304

[15] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L. Berg. 2019. TVQA: Localized, Compositional Video Question Answering. arXiv:1809.01696 [cs.CL] https://arxiv.org/abs/1809.01696

[16] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. arXiv:1910.13461 [cs.CL] https://arxiv.org/abs/1910.13461

[17] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 7871–7880. doi:10.18653/v1/2020.acl-main.703

[18] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv:2005.11401 [cs.CL] https://arxiv.org/abs/2005.11401

[19] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. arXiv:2301.12597 [cs.CV] https://arxiv.org/abs/2301.12597

[20] Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*. Association for Computational Linguistics, Barcelona, Spain, 74–81. https://aclanthology.org/W04-1013

[21] Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022. TAPEX: Table Pre-training via Learning a Neural SQL Executor. arXiv:2107.07653 [cs.CL] https://arxiv.org/abs/2107.07653

[22] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. MTEB: Massive Text Embedding Benchmark. arXiv:2210.07316 [cs.CL] https://arxiv.org/abs/2210.07316

[23] OpenAI. 2024. Hello GPT-4o. https://openai.com/index/hello-gpt-4o/ Accessed: 2024-09-13.

[24] Martin Rader and Klaus Schoeffmann. 2024. lifeXplore at the Lifelog Search Challenge 2024. In *Proceedings of the 7th Annual ACM Workshop on the Lifelog Search Challenge* (Phuket, Thailand) *(LSC '24)*. Association for Computing Machinery, New York, NY, USA, 64–69. doi:10.1145/3643489.3661123

[25] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* 21, 1, Article 140 (Jan. 2020), 67 pages.

[26] Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (April 2009), 333–389. doi:10.1561/1500000019

[27] Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering. *Transactions of the Association for Computational Linguistics* 11 (01 2023), 1–17. doi:10.1162/tacl_a_00530 arXiv:https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00530/2067834/tacl_a_00530.pdf

[28] Wang-Chiew Tan, Jane Dwivedi-Yu, Yuliang Li, Lambert Mathias, Marzieh Saeidi, Jing Nathan Yan, and Alon Halevy. 2023. TimelineQA: A Benchmark for Question Answering over Timelines. In *Findings of the Association for Computational Linguistics: ACL 2023*, Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (Eds.). Association for Computational Linguistics, Toronto, Canada, 77–91. doi:10.18653/v1/2023.findings-acl.6

[29] Ly-Duyen Tran, Thanh Cong Ho, Lan Anh Pham, Binh Nguyen, Cathal Gurrin, and Liting Zhou. 2022. LLQA-lifelog question answering dataset. In *International Conference on Multimedia Modeling*. Springer, 217–228.

[30] Ly Duyen Tran, Thanh-Binh Nguyen, Cathal Gurrin, and Liting Zhou. 2024. MyEachtraX: Lifelog Question Answering on Mobile. In *Proceedings of the 7th Annual ACM Workshop on the Lifelog Search Challenge* (Phuket, Thailand) *(LSC '24)*. Association for Computing Machinery, New York, NY, USA, 93–98. doi:10.1145/3643489.3661128

[31] Ly-Duyen Tran, Liting Zhou, Binh Nguyen, and Cathal Gurrin. 2024. Interactive Question Answering for Multimodal Lifelog Retrieval. In *International Conference on Multimedia Modeling*. Springer, 68–81.

[32] Quang-Linh Tran, Binh Nguyen, Gareth J. F. Jones, and Cathal Gurrin. 2024. MemoriEase 2.0: A Conversational Lifelog Retrieve System for LSC'24. In *Proceedings of the 7th Annual ACM Workshop on the Lifelog Search Challenge* (Phuket, Thailand) *(LSC '24)*. Association for Computing Machinery, New York, NY, USA, 12–17. doi:10.1145/3643489.3661114

[33] Quang-Linh Tran, Binh Nguyen, Gareth J. F. Jones, and Cathal Gurrin. 2024. MemoriQA: A Question-Answering Lifelog Dataset. In *Proceedings of the 1st ACM Workshop on AI-Powered Q&A Systems for Multimedia* (Phuket, Thailand) *(AIQAM '24)*. Association for Computing Machinery, New York, NY, USA, 7–12. doi:10.1145/3643479.3662050

[34] Quang-Linh Tran, Ly-Duyen Tran, Binh Nguyen, and Cathal Gurrin. 2023. MemoriEase: An Interactive Lifelog Retrieval System for LSC'23. In *Proceedings of the 6th Annual ACM Lifelog Search Challenge* (Thessaloniki, Greece) *(LSC '23)*. Association for Computing Machinery, New York, NY, USA, 30–35. doi:10.1145/3592573.3593101

[35] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent Abilities of Large Language Models. arXiv:2206.07682 [cs.CL] https://arxiv.org/abs/2206.07682

[36] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903 [cs.CL] https://arxiv.org/abs/2201.11903

[37] Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2024. InstructRAG: Instructing Retrieval-Augmented Generation with Explicit Denoising. *arXiv preprint arXiv:2406.13629* (2024).

[38] Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking Retrieval-Augmented Generation for Medicine. In *Findings of the Association for*

*Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 6233–6251. doi:10.18653/v1/2024.findings-acl.372

[39] Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking Retrieval-Augmented Generation for Medicine. arXiv:2402.13178 [cs.CL] https://arxiv.org/abs/2402.13178

[40] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600* (2018).

[41] Antonio Jimeno Yepes, Yao You, Jan Milczek, Sebastian Laverde, and Renyu Li. 2024. Financial report chunking for effective retrieval augmented generation. *arXiv preprint arXiv:2402.05131* (2024).

[42] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. arXiv:1904.09675 [cs.CL] https://arxiv.org/abs/1904.09675

[43] Liting Zhou, Cathal Gurrin, Duc-Tien Dang-Nguyen, Graham Healy, Chenyang Lyu, Tianbo Ji, Longyue Wang, Joho Hideo, Ly-Duyen Tran, and Naushad Alam. 2023. Overview of the NTCIR-17 Lifelog-5 Task. In *Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies (NTCIR-17). Tokyo, Japan.*