NTCIR 17 Conference: Proceedings of the 17th NTCIR Conference on Evaluation of Information Access Technologies, December 12-15, 2023, Tokyo, Japan
DOI: https://doi.org/10.20736/0002001334

Automatic Sub-Task Focus: LifeInsight's Contribution to NTCIR-17 Lifelog-5

Thang-Long Nguyen-Ho
University of Science, VNU-HCM
Vietnam National University
Ho Chi Minh City
Vietnam

Tien-Thanh Nguyen-Dang
University of Science, VNU-HCM
Vietnam National University
Ho Chi Minh City
Vietnam

Gia-Huy Vuong
University of Science, VNU-HCM
Vietnam National University
Ho Chi Minh City
Vietnam

Van-Son Ho
University of Science, VNU-HCM
Vietnam National University
Ho Chi Minh City
Vietnam

Xuan-Dang Thai
University of Science, VNU-HCM
Vietnam National University
Ho Chi Minh City
Vietnam

Minh-Khoi Pham
Dublin City University
Ireland

Tu-Khiem Le
Dublin City University
Ireland

Van-Tu Ninh
Dublin City University
Ireland

Minh-Triet Tran
University of Science, VNU-HCM
Vietnam National University
Ho Chi Minh City
Vietnam

ABSTRACT

As the demand for personalized data retrieval systems continues to grow, recent research has emphasized the development of lifelog retrieval mechanisms. Many new research and methods have focused on studying the integration of user interactions and feedback into search engines. In this paper, we introduce the automation approach of LifeInsight, a retrieval system designed explicitly for the NTCIR-17 Lifelog-5 Automatic Task, facilitating a seamless search experience and efficient data mining. Our method entails a two-fold process, where we first enrich the metadata from the raw query, followed by the composition of the retrieval method from input entities. Our proposed system not only enhances the search process but also ensures a comprehensive and detailed analysis of lifelog data for diverse applications. By focusing primarily on the automatic sub-task, we demonstrate the efficacy of our LifeInsight retrieval algorithm, showcasing competitive results that rival those of an expert user.

KEYWORDS

lifelog, quantified-self data, automatic retrieval system

TEAM NAME

HCMUS

SUBTASKS

Lifelog Semantic Access Task (LSAT)

1 INTRODUCTION

The advance of wearable sensors and mobile devices could now enable seamless data recording during daily life. This results in a large-scale multimodal personal data collection, also referred to as the lifelog data [12], spanning different types of data, including images, videos, locations, activities, etc. By analyzing this personal data collection, many potential applications could be proposed to both support humans in daily activities and provide insights into an individual's life. One of the popular applications usually proposed by harnessing the lifelog data is to develop a memory prosthetic that could retrieve the desired life events based on what remains in an individual's memory about these moments. It is also the main focus of the Lifelog Semantic Access Task (LSAT) that has been proposed in the NTCIR-Lifelog challenges for four years [9–11, 30] which motivates the research community to search for the most optimal solutions for the lifelog moment retrieval problem.

The fifth NTCIR-17 Lifelog [31] continues to push the boundaries of lifelog moment retrieval task by introducing a new large-scale lifelog dataset, which spans up to three years of seamless data recording. There are 41 queries with two query types (ad-hoc and known-item search) for two main sub-tasks in the NTCIR-17, which are the automatic sub-task and the interactive one. While the interactive sub-task aims to evaluate the retrieval efficiency and the usability of the interactive lifelog retrieval systems, the automatic sub-task focuses on evaluating the performance of the retrieval algorithm used to solve the queries without the human-in-the-loop. For the interactive sub-task, the experiment configuration stated by the organizers allows the participants to spend five minutes at most for each query. In total, it would take approximately 3 hours and 40 minutes to solve all the queries in this NTCIR-17 Lifelog, while it would take less than three minutes to find the desired moments without human evaluation. Therefore, in this paper, we concentrate on the automatic sub-task to evaluate the performance of the LifeInsight retrieval algorithm with some novel proposed improvements. Our automated methodology has outperformed existing automated processes, delivering competitive results comparable to those of an expert user. Our main contributions to the enhancement of the lifelog moment retrieval system without human-in-the-loop are:

(1) **Enriching Metadata: An Observation Expansion method**. Our approach initiates with an observation expansion process, wherein we leverage both traditional and potential techniques to enrich the metadata and extract nuanced information from the raw query. Employing Semantic Role Labeling (SRL), our system autonomously extracts pertinent entities, enhancing the understanding and depth of the lifelog data. Additionally, we integrate the Large Language Model (LLM) to extract temporal events, irrespective of specific time references, ensuring a comprehensive and contextually rich analysis. The generated comprehensive prompts and their variants further enhance the depth and breadth of the prompt generation process, contributing to a more holistic retrieval process.
(2) **Automatic Composer: Integrating Input Entities for Efficient Retrieval**. Building upon the enriched metadata, our system presents a general composition method that utilizes input entities to formulate a precise and effective query. By generating queries from all available entities and their respective prompts, we employ a preliminary heuristic to ensure the formulation of accurate search queries. This ensures that the lifelog data retrieval process is not only efficient but also comprehensive, catering to a wide range of user needs and requirements.

2 RELATED WORK

The automatic retrieval of lifelog data has garnered significant attention in recent years, aiming to facilitate efficient information extraction from personal multimedia databases. In this section, we discuss prior research efforts that have contributed to the development of advanced systems for lifelog data retrieval, with a particular focus on approaches, enriching metadata, and composing queries from input entities.

There is a vast development in creating competitions to foster the development of interactive retrieval systems. Prime examples of such challenges are the NTCIR-Lifelog task[14], Lifelog Search Challenge [13] and ImageCLEF lifelog[6]. Various methodologies have been documented through the years, encompassing the utilization of diverse vector databases for storage and retrieval, for example, Lifegraph [24] and LifeConcept [3]. These frameworks have effectively leveraged knowledge graphs and concept recommendation methods to streamline the retrieval process by establishing intricate connections between pertinent concepts and associated images.

Recently, an emerging trend involves the integration of vision-language pre-trained models, with particular emphasis on the CLIP model [23]. Noteworthy implementations of this approach include LifeSeeker 4.0 [20], E-Myscéal [28], Memento 2.0 [1], FIRST 3.0 [16], and Voxento [2]. Evaluating the outcomes of methodologies employing vector databases, a significant enhancement in zero-shot image-text retrieval performance has been observed compared to their earlier versions.

Furthermore, there are advanced systems focused on visual content analysis through the creation of extensive meta-data sets, such as lifeXplore [17], PhotoCube [26], and LifeMon [8]. These systems utilize technologies like YOLOv4 [5] and other traditional object detectors based on convolutional neural networks (CNNs).

In addition, there are groundbreaking initiatives in the field of visual content analysis that involve the generation of extensive metadata. Projects like lifeXplore [17], PhotoCube [26], and LifeMon [8] have employed convolutional neural networks (CNNs), including YOLOv4 [5], alongside traditional object detectors. On another note, the pioneering system FIRST [15] has spearheaded the development of context-based suggestions using the Large Language Model (LLM) across various scenarios.

There are many research works in Natural Language Processing (NLP) related to analyzing natural queries in lifelogging. For instance, Semantic Role Labeling helps highlight specific items and has emerged as a promising method for pinpointing entities, making searches more focused. Additionally, large language models using zero-shot learning can help with tasks like Named Entity Recognition (NER) and Information Extraction (IE). Lou et al. [19] suggested a new way to tackle IE tasks, while GOLLIE [25] envisioned IE tasks through the lens of a programming schema language model, outperforming others in this areas.

3 LIFEINSIGHT AT THE NTCIR-17 LIFELOG-5

Unlike the system in the usual interactive track, our approach for automating the process relies entirely on mirroring the typical search behavior of a user within the system. This involves the explicit incorporation of features and the translation of implicit, intuitive actions into precise system actions. Consequently, our paper introduces two primary contributions to observation expansion. Firstly, we present a method for extracting semantic content to facilitate named entity recognition, thereby serving as a weighting filter. Secondly, we propose a context-based prompt generation mechanism inspired by FIRST [15], aiming to enhance the diversity of search queries by generating potential contexts for the description. Additionally, we proposed a method for generating search queries from multiple pieces of information and combining them to yield a final result.

3.1 System Overview for Automatic

Our automated flow is designed to be an extension of LifeInsight interactive version. It uses the results of multiple predefined strategies and generative strategies to produce the final results. The results are refined over numerous rounds of search, revision, and query generation. Figure 1 illustrates the entire search process.

The interactive system [21] that we have built before acts as a **Search Interface Module** in our search flow. This module serves as an interface to convert composed queries into database query language. A concise overview of our system is provided in Section 3.2.

Our system is composed of two key elements. The initial component, known as the **Observation Interpreter**, is responsible for extracting potential facts and creating different prompts from the provided query. It comprises two sub-components: a metadata extractor that identifies relevant entities and terms and a prompt generator that diversifies the input prompts. These components play a critical role in the analysis, providing informative observations that enable the handling of a vast array of comprehensive information. Further insights into this process will be outlined in Section 3.3 and Section 3.4, respectively.

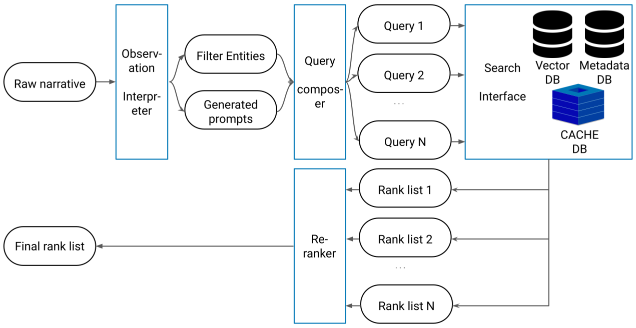

Figure 1: Overview: the narrative undergoes an observation interpretation process, enabling the extraction of vital entities and pertinent contexts. Subsequently, these entities are integrated within a filter alongside various prompt versions and temporal information by the composer, thereby generating queries tailored for the LifeInsight system. Finally, the fuse operation amalgamates results from multiple query iterations to yield an optimal outcome.

The remaining component called **Query composer** consists of multiple algorithms that quantify and analyze the observation, which is then passed after generating variants and extracting pieces of information from the raw query. This helps to construct new queries with more feasible contexts and then merge all the return lists as a final answer. More details about this process will be presented in Section 3.5.

3.2 Search Interface

The search interface module helps translate the input query into database query language, whereas the query can be text or images. However, in the automatic subtask, the system consistently interprets the input query as text. This text undergoes processing by LifeInsight, which autonomously converts it into conceptual data. This encompasses various elements such as location, object, transportation, and color. The text may also contain temporal data, which can be segmented and utilized via a temporal query mechanism to enhance LifeInsight's performance.

Following this, these semantic particulars are converted into visual and semantic vector embeddings through computing engines. The visual embeddings are subsequently inputted into the Milvus[29] vector database for search operations, while the concepts aid in filtering through Elasticsearch[7].

At the end, the module returns top-k relevant images that have the highest score of matching the input original query.

3.3 Metadata Extractor

Filters are essential components of many search systems, allowing users to narrow down their results by specific criteria. As they help to understand the user's intent, filters are the first component to be focused on when automating the search process. In the choosing filter stage, users have specific intentions, such as finding results related to a particular location, time, or objects that could appear in the scene. This could be considered a named entity recognition (NER) problem, where the goal is to identify the correct term for each field of the query's informative subject.

To automatically identify entities in a lifelog query prompt, we propose a highly adaptive method that combines traditional NLP processing and LLM models to zero-shot learn self-defined labels. This approach is motivated by the fact that conventional NLP models do not adapt quickly to new entities, while LLM models are more flexible and can learn new concepts from a few examples. Figure 2 describes the detailed process of our proposed method.

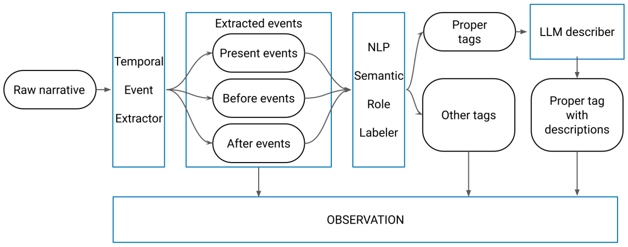

Figure 2: Initially, all candidate events are classified into temporal relationship labels (including before, present, and after) using an LLM model. Then, candidate entities are extracted from each event using a traditional NLP model. Additionally, for each proper noun identified in the previous stage, the LLM describes a comprehensive details sentence. Once all the candidate entities have been classified, special entities, like proper noun phrases or brand labels, are processed through the LLM module to articulate the unique features of the logo design. Finally, the labeled and described events are delivered as an output observation.

The proposed method has several advantages over traditional NLP methods. First, it is more adaptive to new entities, as it does not require a large labeled dataset for training. Second, it is more accurate, as it leverages the power of LLM models to learn complex patterns of entities. Furthermore, it can demonstrate some specific terms as an expert user.

3.3.1 NLP-based extraction method. Natural language processing (NLP) techniques such as semantic role labeling (SRL) and part-of-speech (POS) tagging can extract valuable and specific entities from lifelog data based on the position and type of words in a sentence. To take advantage of these NLP techniques, Stanza [22], a library for natural language processing maintained by the Stanford NLP group [1] is used.

The results of our model include 18 labels [2]. Some of the outputs we selected include geographical location, person name, organization name, album name, artwork, date, and location facilities such

[1] https://nlp.stanford.edu/
[2] Stanza document

RAW INPUT: The train was a marklin brand train and the lifelogger bought it at the weekend.

OUTPUT:
the weekend : DATE ; marklin brand train : OBJECT
lifelogger : OBJECT ; train : OBJECT

TRUECASE INPUT: The train was a Marklin brand train and the Lifelogger bought it at the weekend.

OUTPUT:
the weekend : DATE ; Marklin : PRODUCT
Lifelogger : PRODUCT ; train : OBJECT

Table 1: A comparison between the use of true case and non-true case is evident from the case of the keyword "Marklin" being misrecognized. With true casing, "Marklin" has been correctly identified as a PRODUCT, enabling the use of a visual describer for detailing its attributes. The example demonstrates that nearly all the entity words are appropriately tagged as proper nouns with the accurate entity label. Nonetheless, the model encounters difficulty in truecasing "Lifelogger," interpreting it solely as a proper noun. This particular term poses a considerable challenge since "lifelogger" is utilized exclusively by the dataset in the provided query.

as buildings, airports, highways, and bridges. Using an efficient SRL model to extract meaningful entities from the data with a specified concept, we can create filter fields that are more specific and relevant to the lifelog data.

However, NLP models are also susceptible to errors, especially in caseless cases, when influenced by the user's typos. This often happens because the user does not pay attention or does not think it is essential to the analysis model. Example 1 depicts the results of the prediction model in two cases. After using the proper case normalization method, the results are more reasonable.

3.3.2 *LLM-based extraction method.*

Two pivot challenges faced when researching simulating expert user behaviors is the existence of expected behaviors from users that cannot be achieved in common NLP tasks. These include:

(1) Separation of sub-queries from a single query that contains multiple details that do not describe the same scene. This is necessary because search engines are essentially model-centric, meaning they can only understand a captured scene at a specific time. Therefore, users must interpret the scene visually and grasp the essential keywords in the metadata fields. This separation can be based on the time and amount of visual information in the sentence.

(2) Recognition of details of brands, products, and organizations not included in the sentence to refine and interpret those details appropriately. This is necessary because we can have supporting knowledge for some proper names. This depends significantly on how well the user understands the individual phrases in the data. For example, the abbreviation "VNA" for airline tickets can be interpreted as "Vietnam Airlines"; the logo of a global brand could be described with color information or special features; for example, the "Gucci" logo could be demonstrated as a distinctive shape made of two capital G's, which are intertwined.

To address the inquiries above, a large language model (LLM) is used as an expert to focus on significant objects in the description. This enables the LLM to comprehend the implicit meanings of the instruction that many NLP models cannot yet handle, such as temporal relationship extraction and identifying the key events in a query.

INPUT:
The lifelogger was watching the Beatles' rooftop concert on TV (not in music metadata). The lifelogger was at home watching YouTube on TV for about 90 minutes, after doing some computer work at home.

PRESENT:
The lifelogger was watching the Beatles rooftop concert on tv
BEFORE:
The lifelogger was at home doing some computer work
AFTER:
The lifelogger was at home watching YouTube on TV for about 90 minutes

Table 2: Temporal extraction module categorizes input data into distinct temporal classes (before, present, after) and effectively eliminates temporal information from the extracted event representation ("after doing")

To allow the automatic system to describe some details of well-known brands or products based on knowledge, an interaction between an LLM model and an external engine is used to find more information about potential keywords and describe the abbreviations, which contain crucial information. One example of how the logo descriptions contribute to our system is shown in Table 3.

Nevertheless, one of the persistent challenges pertains to effectively parsing the generated outputs into structured language conforming to specific system requirements. In this context, the pioneering work LMQL [4], a novel solution aimed at facilitating the seamless control of token generation and parsing of output results. This framework promises to enhance the overall efficiency and effectiveness of language model-based methods, thereby enabling smoother communication with our system.

3.4 Context-based Prompting

Using subqueries to obtain an overview from multiple perspectives is a crucial step in locating the search answer. This process necessitates some expertise in extracting information from the original [3]prompt to create a new narrative of the object being sought. This

[3] Throughout this section, the term "prompt" refers to a natural language description, while the term "query" denotes a structured message encompassing natural language description, filtering, and temporal structure. This distinction is made to prevent any potential confusion between the two terms.

NARRATIVE: The train was a Marklin brand train and the lifelogger bought it at the weekend. **INPUT**: Marklin
OUTPUT: A bold, red wordmark "Märklin" with a stylized "M" and a red stripe above and below.
NARRATIVE: Moments that show the lifelogger was meeting friends outside a bar called the Brazen Head, before walking to another bar for drinks. **INPUT**: Brazen Head
OUTPUT: The Brazen Head logo in Ireland is a blue shield with a gold harp, surrounded by a gold wreath.

Table 3: LLM describer articulates visual attributes of a logo of a provided brand, facilitating a comprehensive understanding of the logo characteristics through its knowledge base.

could entail extracting a subset of information that refers to an event, employing various query techniques, generalizing details to improve the model's comprehension, or simply selecting key details. An expert user can flexibly apply various skills to find search targets for themselves.

To imitate effective prompt generation, we define the following concepts: global context prompt generation and local context aware prompt generation. We then use these concepts to visualize how to generate new prompts. Finally, we dive deep into how to use filters and the temporal search feature in the finding process.

To generate a diverse and relevant corpus of queries, we propose two concepts: global and local context awareness. Global context awareness considers the context in which a prompt is likely to occur, while local context awareness considers various ways of expressing the same meaning. FIRST [15] approach also identifies two corresponding definitions for prompting: deductive suggestion and abductive suggestion. The difference between prompt generation types shown at 4

3.4.1 Local context awareness prompting. Local context awareness allows us to express information in many different ways. Given a piece of information, there are multiple ways to infer its meaning, while different ways of ordering the concepts or changing the level of detail of the prompt can all have an impact on the result. Our goal is to generate sentences that keep the same meaning and components but create multiple versions of them.

3.4.2 Global context awareness prompting. Global context awareness allows us to find multiple possible cases for a prompt. We often need to make an educated guess and try different scenarios. The large language model (LLM) should try to find the most likely scenario based on its knowledge and the information it can access from multiple sources. It can artfully craft these identified factors into an entirely novel prompt, thereby demonstrating its proficiency in contextual analysis and generative capabilities. This level of context awareness not only enriches the model's problem-solving aptitude but also enhances its versatility in generating contextually relevant and coherent responses.

INPUT PROMPT: It was at home and there were a few people at the BBQ. The lifelogger had driven home before cooking lots of different foods on the BBQ.
GLOBAL CONTEXT PROMPTING: Capturing the cooking process and the interactions at a home BBQ, as the lifelogger prepares a variety of delicious foods.
LOCAL CONTEXT PROMPTING: At home, there were a few individuals at the BBQ. The lifelogger had driven home before cooking many, different dishes on the grill.

Table 4: A comparison between prompt generation. Illustrating the distinctions in prompt generation methodologies. In the case of glocal context prompting, the model introduces a variant, termed 'lifelogger prepares', while in the local context, certain words such as 'BBQ' and 'grill' remain synonymous and unchanged, which contextual sentence coherence is preserved throughout the generation process.

3.5 Query composer

In this part, we proposed a general framework that could mimic the searching techniques of experts. In particular, we focused on the use of generated prompts, which are short phrases or questions that are used to guide a search engine. We also incorporated filters into our search process, which allowed us to refine our results and exclude irrelevant information. Additionally, time awareness was considered in query generation, which allowed the authors to describe their query at a time range, making it less complex for the search engine to understand. By carefully considering all of these factors, we devised a simple and effective search process, illustrated in algorithm 1.

Algorithm 1 Query composition process

```
function GetResults(text)
    $temporal \leftarrow$ temporalExtractor($text$)
    $metadata \leftarrow$ metadataExtractor($temporal$)
    $totalFilter \leftarrow$ refineFilter($metadata$)
    $variants \leftarrow$ generatePrompts($text$)
    $results \leftarrow$ []
    for filter in comb(totalFilter) do
        $results.append$(search($variants, filter$))
    $rankedList \leftarrow$ reRank($results$)
     return rankedList
```

Since there is no ability to interact with the system compared to interactive tasks, the filters generated by the Metadata extractor can be erroneous, or the queried filter may not be present in the dataset's metadata. This can lead to the elimination of query results. To mitigate this, we perform pre-processing to eliminate filters that can potentially return empty results before they are used for searching in the retrieval system. Then, combinations of filters are generated along with generated prompts to create queries that are inputted into the system. Combinations of filters are generated by gradually removing filters in a specific order based on their potential impact on the results. Through experimentation, we have determined the optimal order as follows: location → OCR → time → concept. This approach involves iteratively removing filters and conducting searches until a sufficient amount of information is obtained. It is crucial to strike a balance and avoid removing too many filters, as this can lead to results that do not align with the query description.

The results are aggregated and re-ranked, similar to the AI-assisted crowd-sourcing mode of Query Assistance described in LifeInsight [21]. In short, the formula is:

$$score_x = sim(\vec{q}_0, \vec{e}_x) + \sum_{i=1}^{N} sim(\vec{q}_i, \vec{e}_x)$$

where $\vec{q}_0$ is the embedding vector of the origin query, N is the number of variants generated by LLM, $\vec{q}_i$ is the i^{th} variant query and $\vec{e}_x$ is the embedding vector of image x.

4 EXPERIMENT

4.1 Experimental Setup

To evaluate the performance of our proposed methods, we conducted experiments on two datasets:

- LSC'22: This dataset contains various lifelog data, including images, videos, text, and sensor data. It is a challenging dataset that has been used to develop algorithms for lifelog retrieval and analysis for many years.
- NTCIR-17: This dataset is similar to LSC'22 regarding data types, but it has a different question set. We used this dataset to evaluate the capabilities of our methods on another dataset. This is also the official dataset of the NTCIR-17 benchmark.

4.1.1 *Evaluation Metric.*

We used several metrics to evaluate the performance results of our methods, including:

- **Mean Average Precision (mAP):** Measures the overall average precision across all queries.
- **Precision at K (P@K):** Measures the average precision at the top-K retrieved items for all queries.
- **Recall at K (R@K):** Measures the average recall at the top-K retrieved items for all queries.

4.1.2 *Experimental Procedure.*

We experimented according to the following steps:

(1) Hyperparam tuning: Our system's performance optimization involved the implementation of diverse hyperparameter tuning methodologies and query generation strategies. Additionally, we experimented with different vision-language models, specifically CLIP [27] and BLIP-2 [18]. Leveraging the LSC'22 queries, tailored explicitly for Known Item Search, our primary aim was to ensure precise identification of a single target object, reflecting the nature of the competition. As a result, we adopted R@K as the key evaluation metric for this phase.

(2) NTCIR-17 Dataset Evaluation: We conducted an extensive evaluation of our trained models using the NTCIR 17 dataset. Given the diverse nature of the NTCIR challenge, encompassing both ad-hoc and known item search tasks, the evaluation criteria primarily centered around assessing related items. Consequently, the evaluation standards employed were P@K and mAP.

4.2 Experimental results

Table 6 demonstrates our internally evaluated results on the LSC'22 queries, which consist of a total of 14 Known Item Search queries. We have used the top-k Recall metric for evaluation. Our methodology has achieved near-optimal results at $K = 50$ for the Known Item Search query set, incorporating the contributions above sequentially.

Table 5 presents the performance of our automated process on the NTCIR 17 queries compared to user-generated results. The findings underscore the superior performance of our automated process over that of typical users, showcasing competitive results in the category of mean average precision (mAP) relative to expert users. Notably, our precision metrics, specifically P@5 and P@10, indicate the efficacy of our queries without the need for manual intervention, suggesting that non-expert users employing our system with informed decision-making could potentially attain performance akin to that of an expert. Table 7 shows the full results of all our improvements.

One of the limitation of our approach pertains to the utilization of filters, which seems to yield a dual effect. For instance, in cases "in front of the LIDL store," our system identifies the location as the LIDL store, resulting in the retrieval of images situated inside the store, thus deviating from the user's initial search expectations.

5 CONCLUSION

Our automated search system, built upon a foundation of user behavior mirroring and comprehensive observation expansion, demonstrates an enhanced capability to process complex search queries. By integrating semantic entity extraction, context-based prompt generation, and query composition techniques, our system offers an efficient and intuitive approach to automated search processes. Notably, our system's performance has surpassed most existing automated processes, marking a significant leap forward in the field.

ACKNOWLEDGEMENT

REFERENCES

[1] Naushad Alam, Yvette Graham, and Cathal Gurrin. 2022. Memento 2.0: An Improved Lifelog Search Engine for LSC'22. In *Proceedings of the 5th Annual on Lifelog Search Challenge* (Newark, NJ, USA) *(LSC '22)*. Association for Computing Machinery, New York, NY, USA, 2–7. https://doi.org/10.1145/3512729.3533006

[2] Ahmed Alateeq, Mark Roantree, and Cathal Gurrin. 2022. Voxento 3.0: A Prototype Voice-Controlled Interactive Search Engine for Lifelog. 43–47. https://doi.org/10.1145/3512729.3533009

[3] Wei-Hong Ang, An-Zi Yen, Tai-Te Chu, Hen-Hsen Huang, and Hsin-Hsi Chen. 2021. LifeConcept: An Interactive Approach for Multimodal Lifelog Retrieval through Concept Recommendation *(LSC '21)*. Association for Computing Machinery, New York, NY, USA, 47–51. https://doi.org/10.1145/3463948.3469070

[4] Luca Beurer-Kellner, Marc Fischer, and Martin Vechev. 2023. Prompting Is Programming: A Query Language for Large Language Models. *Proc. ACM Program. Lang.* 7, PLDI, Article 186 (jun 2023), 24 pages. https://doi.org/10.1145/3591300

Table 5: Results of LifeInsight's runs in NTCIR-17 queries. "Ux - Ay" stands for "User x using approach y". More detail in our interactive paper.

Run	# queries attempted	num_q	# images correct/submitted	mAP	P@5	P@10
Best Auto	41	41	**751**/3892	**0.2924**	**0.4098**	**0.3561**
Expert	40	41	350/932	0.1640	0.3200	0.2450
U1-A1	40	41	213/951	0.0614	0.2150	0.1725
U2-A1	37	41	226/1221	0.1687	0.2162	0.1676
U3-A1	36	41	374/1181	0.1270	0.3056	0.2528
U4-A1	38	41	282/1024	0.1255	0.2526	0.1974
U1-A2	35	41	291/1285	0.1290	0.2514	0.2286
U2-A2	36	41	182/1121	0.0872	0.2056	0.1611
U3-A2	39	41	208/870	0.1276	0.2923	0.2308

Table 6: R@K for LSC'22 queries. All results of each component are added to the system (SEMANTIC: semantic search, REPHRASE: using generated prompts, SRL: extracting metadata using SRL, LLM: extracting metadata using LLM, TEMPORAL: applying temporal search).

Run	R@1	R@3	R@5	R@10	R@50	R@100
CLIP_SEMANTIC	0.4286	0.5714	0.7875	0.7875	0.8571	0.8571
CLIP_SEMANTIC_FILTER_SRL	0.5000	0.6429	0.7857	0.8571	0.8571	0.8571
CLIP_SEMANTIC_FILTER_SRL_LLM	0.5714	0.7143	0.8571	0.8571	0.9286	0.9286
CLIP_REPHRASE_FILTER_SRL_LLM	0.7143	0.7857	0.8571	0.8571	0.9286	0.9286
CLIP_REPHRASE_FILTER_SRL_LLM_TEMPORAL	**0.7143**	**0.8571**	**0.8571**	**0.9286**	**1.0000**	**1.0000**
BLIP_SEMANTIC	0.2857	0.4286	0.4286	0.5000	0.5714	0.5714
BLIP_SEMANTIC_FILTER_SRL	0.2857	0.5000	0.5000	0.5000	0.6429	0.6429
BLIP_SEMANTIC_FILTER_SRL_LLM	0.4286	0.5000	0.6429	0.6429	0.6429	0.6429
BLIP_REPHRASE_FILTER_SRL_LLM	0.4286	0.5000	0.6429	0.7143	0.7143	0.7143
BLIP_REPHRASE_FILTER_SRL_LLM_TEMPORAL	0.5714	0.7143	0.7143	0.7143	0.7857	0.8571

Table 7: Results of Automatic LifeInsight's runs in NTCIR-17. All results of each component are added to the system (SEMANTIC: semantic search, REPHRASE: using generated prompts, SRL: extracting metadata using SRL, LLM: extracting metadata using LLM, TEMPORAL: applying temporal search).

Run	# queries attempted	# images correct/submitted	mAP	P@5	P@10	P@100
CLIP_SEMANTIC	41	569/4003	0.2055	0.3366	0.2683	0.1388
CLIP_SEMANTIC_FILTER_SRL	41	560/3475	0.2174	0.3659	0.2976	0.1366
CLIP_SEMANTIC_FILTER_SRL_LLM	41	595/3467	0.2136	0.3756	0.3098	0.1451
CLIP_REPHRASE_FILTER_SRL_LLM	41	641/3649	0.2679	0.3756	0.3171	0.1563
CLIP_REPHRASE_FILTER_SRL_LLM_TEMPORAL	41	**751**/3892	**0.2924**	**0.4098**	**0.3561**	**0.1832**
BLIP_SEMANTIC		574/4000	0.1560	0.3073	0.2659	0.1400
BLIP_SEMANTIC_FILTER_SRL	41	497/3529	0.1626	0.3171	0.2707	0.1212
BLIP_SEMANTIC_FILTER_SRL_LLM	41	593/3556	0.1647	0.3366	0.2927	0.1446
BLIP_REPHRASE_FILTER_SRL_LLM	41	567/3609	0.1713	0.3366	0.2732	0.1383
BLIP_REPHRASE_FILTER_SRL_LLM_TEMPORAL	41	530/3780	0.1882	0.3268	0.2585	0.1293

[5] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv:2004.10934 [cs.CV]

[6] Duc-Tien Dang-Nguyen, Luca Piras, Michael Riegler, Minh-Triet Tran, Liting Zhou, Mathias Lux, Tu-Khiem Le, Van-Tu Ninh, and Cathal Gurrin. 2019. Overview of ImageCLEFlifelog 2019: Solve my life puzzle and Lifelog Moment Retrieval. In *CLEF2019 Working Notes (CEUR Workshop Proceedings)*. CEUR-WS.org <http://ceur-ws.org>, Lugano, Switzerland.

[7] elasticsearch. 2015. *elasticsearch/elasticsearch*. https://github.com/elasticsearch/elasticsearch

[8] Alexander Faisst and Björn Jónsson. 2021. LifeMon: A MongoDB-Based Lifelog Retrieval Prototype. 75–80. https://doi.org/10.1145/3463948.3469066

[9] Cathal Gurrin, Hideo Joho, Frank Hopfgartner, Liting Zhou, and Rami Albatal. 2016. Overview of NTCIR-12 Lifelog Task. In *NTCIR Conference on Evaluation of Information Access Technologies*. https://api.semanticscholar.org/CorpusID:10169871

[10] Cathal Gurrin, Hideo Joho, Frank Hopfgartner, Liting Zhou, Duc-Tien Dang-Nguyen, Rashmi Gupta, and Rami Albatal. 2017. Overview of NTCIR-13 Lifelog-2 Task. In *NTCIR Conference on Evaluation of Information Access Technologies*. https://api.semanticscholar.org/CorpusID:46168176

[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. https://api.semanticscholar.org/CorpusID:197644324

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

[13] Cathal Gurrin, Björn Þór Jónsson, Duc Tien Dang Nguyen, Graham Healy, Jakub Lokoc, Liting Zhou, Luca Rossetto, Minh-Triet Tran, Wolfgang Hürst, Werner Bailer, and Klaus Schoeffmann. 2023. Introduction to the Sixth Annual Lifelog Search Challenge, LSC'23. In *Proc. International Conference on Multimedia Retrieval (ICMR'23)* (Thessaloniki, Greece) *(ICMR '23)*. New York, NY, USA. https://doi.org/10.1145/3591106.3592304

[14] Gurrin, Cathal and Hopfgartner, Frank and Dang-Nguyen, Duc-Tien and Nguyen, Thanh-Binh and Healy, Graham, and Albatal, Rami, and Zhou, Liting. 2022. Overview of the NTCIR-16 Lifelog-4 Task. In *Proceedings of the 16th NTCIR Conference on Evaluation of Information Access Technologies (NTCIR-16)*. Tokyo, Japan.

[15] 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* (Thessaloniki, Greece) *(LSC '23)*. Association for Computing Machinery, New York, NY, USA, 47–52. https://doi.org/10.1145/3592573.3593104

[16] Nhat Hoang-Xuan, Hoang-Phuc Trang-Trung, E.-Ro Nguyen, Thanh-Cong Le, Mai-Khiem Tran, Tu-Khiem Le, Van-Tu Ninh, Cathal Gurrin, and Minh-Triet Tran. 2022. Flexible Interactive Retrieval SysTem 3.0 for Visual Lifelog Exploration at LSC 2022. In *LSC@ICMR 2022: Proceedings of the 5th Annual on Lifelog Search Challenge, Newark, NJ, USA, June 27 - 30, 2022*, Cathal Gurrin, Graham Healy, Liting Zhou, Björn Þór Jónsson, Duc-Tien Dang-Nguyen, Jakub Lokoc, Minh-Triet Tran, Wolfgang Hürst, Luca Rossetto, and Klaus Schoeffmann (Eds.). ACM, 20–26. https://doi.org/10.1145/3512729.3533013

[17] Andreas Leibetseder and Klaus Schoeffmann. 2021. lifeXplore at the Lifelog Search Challenge 2021. 23–28. https://doi.org/10.1145/3463948.3469060

[18] 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]

[19] Jie Lou, Yaojie Lu, Dai Dai, Wei Jia, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2023. Universal Information Extraction as Unified Semantic Matching. *Proceedings of the AAAI Conference on Artificial Intelligence* 37, 11 (Jun. 2023), 13318–13326. https://doi.org/10.1609/aaai.v37i11.26563

[20] Thao-Nhu Nguyen, Tu-Khiem Le, Van-Tu Ninh, Minh-Triet Tran, Thanh Binh Nguyen, Graham Healy, Sinéad Smyth, Annalina Caputo, and Cathal Gurrin. 2022. LifeSeeker 4.0: An Interactive Lifelog Search Engine for LSC'22. In *Proceedings of the 5th Annual on Lifelog Search Challenge* (Newark, NJ, USA) *(LSC '22)*. Association for Computing Machinery, New York, NY, USA, 14–19. https://doi.org/10.1145/3512729.3533014

[21] 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* (Thessaloniki, Greece) *(LSC '23)*. Association for Computing Machinery, New York, NY, USA, 59–64. https://doi.org/10.1145/3592573.3593106

[22] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. arXiv:2003.07082 [cs.CL]

[23] 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.CV]

[24] Luca Rossetto, Matthias Baumgartner, Ralph Gasser, Lucien Heitz, Ruijie Wang, and Abraham Bernstein. 2021. Exploring Graph-Querying Approaches in LifeGraph. In *Proceedings of the 4th Annual on Lifelog Search Challenge* (Taipei, Taiwan) *(LSC '21)*. Association for Computing Machinery, New York, NY, USA, 7–10. https://doi.org/10.1145/3463948.3469068

[25] Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2023. GoLLIE: Annotation Guidelines improve Zero-Shot Information-Extraction. arXiv:2310.03668 [cs.CL]

[26] Jihye Shin, Alexandra Waldau, Aaron Duane, and Björn Jónsson. 2021. PhotoCube at the Lifelog Search Challenge 2021. 59–63. https://doi.org/10.1145/3463948.3469073

[27] 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*. 23–28.

[28] Tran, Ly-Duyen and Nguyen, Manh-Duy and Nguyen, Binh and Lee, Hyowon and Zhou, Liting and Gurrin, Cathal. 2022. E-Myscéal: Embedding-Based Interactive Lifelog Retrieval System for LSC'22. In *Proceedings of the 5th Annual on Lifelog Search Challenge* (Newark, NJ, USA) *(LSC '22)*. Association for Computing Machinery, New York, NY, USA, 32–37. https://doi.org/10.1145/3512729.3533012

[29] Jianguo Wang, Xiaomeng Yi, Rentong Guo, Hai Jin, Peng Xu, Shengjun Li, Xiangyu Wang, Xiangzhou Guo, Chengming Li, Xiaohai Xu, Kun Yu, Yuxing Yuan, Yinghao Zou, Jiquan Long, Yudong Cai, Zhenxiang Li, Zhifeng Zhang, Yihua Mo, Jun Gu, Ruiyi Jiang, Yi Wei, and Charles Xie. 2021. Milvus: A Purpose-Built Vector Data Management System. In *Proceedings of the 2021 International Conference on Management of Data* (Virtual Event, China) *(SIGMOD '21)*. Association for Computing Machinery, New York, NY, USA, 2614–2627. https://doi.org/10.1145/3448016.3457550

[30] Zhou, Cathal Gurrin, and Graham Healy. [n. d.]. Overview of the NTCIR-16 Lifelog-4 Task. https://api.semanticscholar.org/CorpusID:251395683

[31] Zhou, Liting and Gurrin, Cathal and Dang-Nguyen, Duc-Tien and Healy, Graham, and Lyu, ChenYang and Ji, Tianbo and Wang, Longyue and Hideo Joho and Tran, Ly-Duyen 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.