An Interactive Lifelog Search Engine for LSC2018

Adrià Alsina and Xavier Giró
Universitat Politècnica de Catalunya
Barcelona
Spain
Cathal Gurrin
Insight Centre for Data Analytics
Dublin City University
Dublin, Ireland

Conference'17, July 2017, Washington, DC, USA

Adrià Alsina and Xavier Giró
Universitat Politècnica de Catalunya
Barcelona
Spain
Cathal Gurrin
Insight Centre for Data Analytics
Dublin City University
Dublin, Ireland

1 INTRODUCTION

Lifelogging is a new concept and has only recently attracted the attention of the research community [3]. One standard definition of lifelogging states that it is *"the action or practice of making a continuous record of one's daily activities by means of a digital device or computer application"*. The growing research interest in lifelogging has been facilitated by the market-availability of a range of lifelogging and quantified self devices that can digitally capture a wide range of life activities, from wearable cameras for all-of-lifelogging to the more targeted health and wellness devices commonly referred to as quantified self devices [5].

With such a range of lifelogging devices, one can easily gather, in an automated manner, a wide range of data about the life experience of the individual, such as image data from smartphones or wearable cameras, audio recordings, location data, biometric data, to name only a few.

The increase of interest in lifelogging has resulted in many different research challenges being developed, such as the NTCIR lifelog task Semantic Access Task (LSAT) of the the NTCIR-12 challenge [2] and the ImageCLEF [4] lifelog task. This LSC lifelog task posed unique challenges due to the interactive nature of the task, hence we felt that our retrieval approaches would be suitable for this task. The interactive nature of the challenge means that any system developed for the LSC would need to be optimised to select or locate relevant content from a comparatively small archive of lifelog data, which in this case, consisted of multimedia data from wearable cameras, biometric data from smartwatches, human activity data

from smartphones and computer usage data to identify information access actitivies.

Without there having been similar tasks in the past, there are not many interactive retrieval engines developed for lifelog data that we could refer to. One system of note is the LEMoRe [1], an interactive lifelog retrieval engine, developed in the context of the Lifelog Semantic Access Task (LSAT) of the the NTCIR-12 challenge [2]. LEMoRe integrated classical image descriptors with high-level semantic concepts extracted by Convolutional Neural Networks [6], powered by a graphical user interface that uses natural language processing to process a user's query.

Building on experiences from this work, we have developed the experimental serach engine which forms the contribution of this paper. The general approach taken is to develop an indexing and filtering tool, utilising an SQL database, with a is based on the development of a lifelogging image browser that works with a full text search inside a SQL database.

2 A FOUR-STEP APPROACH TO INTERACTIVE LIFELOG RETRIEVAL

In designing our lifelog retrieval engine, we separate the task into four challenges:

- Storage and Indexing of the source data. Lifelogging typically involves multiple sensors, as did LSC dataset. The first task is to choose, process and align these data files so that they are temporally arranged.
- Development of a multi-faceted query interface. The multimodal nature of the LSC dataset naturally lends itself to multi-faceted query generation, so we have develop a query mechanism that allows a user to enter queries based on many facets, such as visual concept, date/time and place.
- For retrieving all the data is important to show at the beginning the images that can interest more to the user. Fir this reason is important to use a good search engine that orders the images depending on the input query.
- Presentation of a Ranked List from a lifelog dataset. Given the temporal nature of lifelog data, it is sensible to consider how to exploit this temporal data when developing a ranked list.

We will now explain how we solved each of these four challenges.

2.1 Storage and Indexing

LSC lifelog dataset consists of a wide array of multimodal data of an individual's life experience captured via wearable and software sensors. The dataset was gathered over a period of 27 days and it includes:

- Multimedia data : Represents about 1,500 images per day taken by a wearable camera. So, if we have 27 days of data it means that we have a database of images about 4,500 images. These images are accompanied by a concept detector file where we can find the top concept main words for every image. Also, multimedia data is formed for a music listing history.
- Biometric data : It is formed for biometric data like heart rate, galvanic skin response, calorie burn and steps stored almost every minute of the day. Also we have data about blood pressure and blood sugar daily.
- Human activity data : Consists in information about the semantic locations visited, the physical activities, the daily mood and a diet log made of manual logging of photos.
- Computer Usage : Represents a vector of every minute stored data filtered using blacklist, anonymised and then stemmed.

As more small is the database as better it is. Having a lot of information means slower processes. For this reason the first step is to think and select the information that is useful for us. Of all this data we select to work with the images and the associated concept data including the time that every image was taken. And also we select to work with almost every minute heart rate, location and activity data.

Once we know which data is interesting for us we have to store all this data together in a database for make the search retrieval easier. For this case we select to store all this data in a different column of a MySQL database. MySQL is an open source relational database management system based on SQL language. We select to work with MySQL because it allows us to make full text search queries in our database.

2.2 Multi-faceted query interface

For letting the user introduce the queries we create and easy and easy functional interface. The design of this interface have two main parts as we can see in the *Figure 2.* In the left side we can see the zone where the user is be able to introduce the queries. We create an easy way to introduce the queries. It consists in one search bar where the user can introduce main words. One selector for the day of the week and for an exact day in the calendar. Also we introduce a range selector for the heart rate level. And finally, some check boxes for selecting some other things. The other part, in the right, is where the images are displayed, at the beginning it is completely empty as we can see in the *Figure 2* and once we make the query it fills with the images.

Also this interface is created in two different part. First of all, all the design part is created using a library of JavaScript called React.js. React.js has recently came as one of the most popular JavaScript libraries for creating user interfaces, it is actually used by most of the main web sites like Facebook or Instagram, because of its speed, simplicity, and scalability. The second part was is the deign of our server. The server is the one that receives the queries
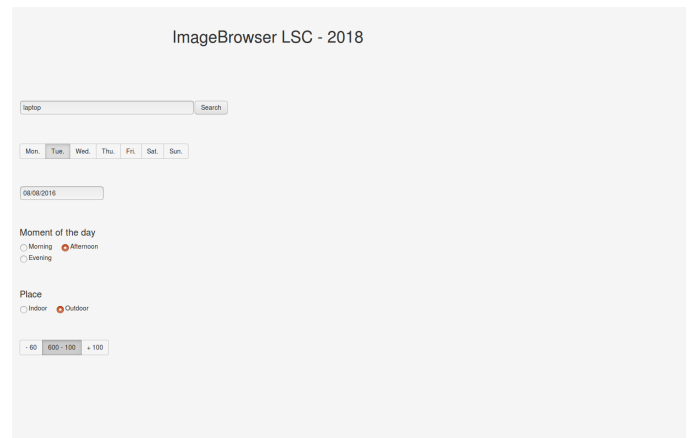

Figure 1: Design of the query interface.

of JavaScript and process them. Once processed is also the one that interacts with MySQL database and finally gives a response to our main page displaying the images. This server is deign using the Python library Flask. Flask is a web framework that give us simplicity, flexibility and fine-grained control. Flask is also very popular in the web deign community,is used for important websites like LinkedIn or Pinterest.

2.3 Ranking Engine

Here we need more information... What is the ranking algorithm. How does it work? What is the indexable unit? I assume it is the image, with associated medatata.. .then we need to clearly say it

Naturally, this basic data access process must be enhanced to support the generation of ranked list of relevant content for any given query. This is the main reasons why we choose to work with MySQL. Every row of the database represents a different image, and every column a different kind of data that we are interested on search. With MySQL we can do a full text search queries in this database. Full text search text give us the tool to make queries to the database and order the result depending on the score. This score is placed using a TF-IDF algorithm. TF-IDF is a numerical statistic that is intended to reflect how important a word is in our database, so the words that appear less in our database have a higher punctuation than the once that appear more. So, when we introduce one query with two words, like laptop and horse. The word horse will have a better score than the word laptop that appears a lot of times in the database. So if in an image appears a horse and a laptop the punctuation of both words will be added and if just appears one of these words the sore will be just the one of this word. So, even if one of the parameters that the user introduces doesn't match with the image these will be displayed in the page but in last rows giving less importance to them. Also MySQL algorithm works with a stopword handing. That means that our search engine avoid irrelevant results caused by matching common words such as *a* and *the*.

This ranked list is merged with the filters provided by the other data facets, such as time of day, day of week, or location. Such a faceted filtering model is commonly employed for interactive

Adrià Alsina and Xavier Giró
Universitat Politècnica de Catalunya
Barcelona
Spain
Cathal Gurrin
Insight Centre for Data Analytics
Dublin City University
Dublin, Ireland

retrieval systems [7]/. Every column means a different filed so we just need to make the queries in the corresponding filed to make our rank engine work perfectly.

2.4 Result List Presentation

Upon submission of a query, the system generates a ranked list of images for display to the user. The images are displayed as you can see in the *Figure 2*. Rather than simply display a ranked list, the system aims to provide some context to the user giving some details about the main image. Also, next to the main image are displayed previous and next images. These images are selected using the ranking engine that brings us MySQL. Once we select the main image we browse in the database the first previous and next images that have a different score respect to the main one. The user is then expected to click on one image submit it to the LSC server for validation. Our expectation is that the image to the left provides contextual information on what the lifelogger was doing previously and the image on the right provides information on the next activities. We include this functionality because the development topics contained some temporal topics that require knowledge about the preceding or succeeding activities of the lifelogger. However, as more extensive is the query as more similar are the images that are next to the main one because a little change on the image means a change in the information and therefore in the score.
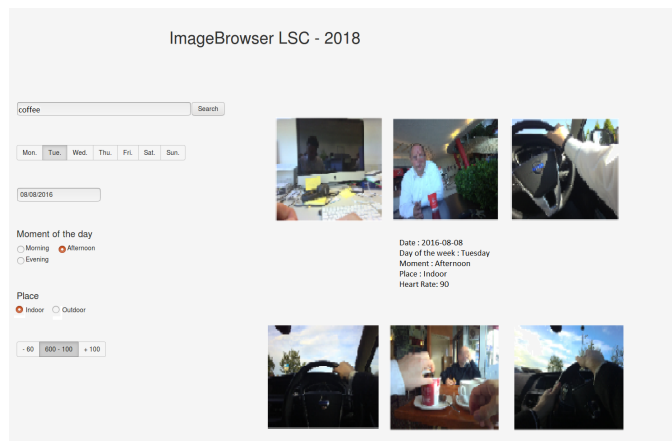

Figure 2: Display of the images.

3 CONCLUSION

In this paper, we describe the data and the task from the LSC 2018 lifelog search challenge. We describe the prototype interactive search engine that we built for the challenge and we presented the interface and extensions to the basic search engine.

ACKNOWLEDGEMENTS

REFERENCES

[1] G. de Oliveira-Barra, X. Giró-i Nieto, A. Cartas-Ayala, and P. Radeva. Lemore: A lifelog engine for moments retrieval at the ntcir-lifelog lsat task. In *The 12th NTCIR Conference, Evaluation of Information Access Technologies*, Tokyo, Japan, 06/2016 2016. National Institute of Informatics (NII), National Institute of Informatics (NII).

[2] C. Gurrin, H. Joho, F. Hopfgartner, L. Zhou, and R. Albatal. Overview of ntcir-12 lifelog task, 2016.

[3] C. Gurrin, A. F. Smeaton, and A. R. Doherty. Lifelogging: Personal big data. *Foundations and Trends in Information Retrieval*, 8(1):1–125, 2014.

[4] B. Ionescu, H. Müller, M. Villegas, H. Arenas, G. Boato, D.-T. Dang-Nguyen, Y. Dicente Cid, C. Eickhoff, A. G. Seco de Herrera, C. Gurrin, B. Islam, V. Kovalev, V. Liauchuk, J. Mothe, L. Piras, M. Riegler, and I. Schwall. Overview of imageclef 2017: Information extraction from images. In G. J. Jones, S. Lawless, J. Gonzalo, L. Kelly, L. Goeuriot, T. Mandl, L. Cappellato, and N. Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 315–337, Cham, 2017. Springer International Publishing.

[5] J. Meyer, S. Simske, K. Siek, C. Gurrin, and H. J. Hermens. Beyond quantified self: data for wellbeing. In *Extended Abstracts on Human Factors in Computing Systems*, 2014.

[6] J. Schmidhuber. Deep learning in neural networks: An overview. *CoRR*, abs/1404.7828, 2014.

[7] D. Tunkelang. Faceted search. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 1(1):1–80, 2009.