LEMoRe: A Lifelog Engine for Moments Retrieval at the NTCIR-Lifelog LSAT Task

Gabriel de Oliveira Barra
University of Barcelona
gabriel.deoliveira@ub.edu

Alejandro Cartas Ayala
University of Barcelona
Funded by CONACyT, Mexico
alejandro.cartas@ub.edu

Marc Bolaños
University of Barcelona
marc.bolanos@ub.edu

Mariella Dimiccoli
Computer Vision Center
mariella.dimiccoli@cvc.uab.es

Xavier Giro-i-Nieto
Polythecnic University of Catalonia
xavier.giro@upc.edu

Petia Radeva
University of Barcelona
petia.ivanova@ub.edu

ABSTRACT

Semantic image retrieval from large amounts of egocentric visual data requires to leverage powerful techniques for filling in the semantic gap. This paper introduces LEMoRe, a Lifelog Engine for Moments Retrieval, developed in the context of the Lifelog Semantic Access Task (LSAT) of the the NTCIR-12 challenge and discusses its performance variation on different trials. LEMoRe integrates classical image descriptors with high-level semantic concepts extracted by Convolutional Neural Networks (CNN), powered by a graphic user interface that uses natural language processing. Although this is just a first attempt towards interactive image retrieval from large egocentric datasets and there is a large room for improvement of the system components and the user interface, the structure of the system itself and the way the single components cooperate are very promising.

Team Name

LEMoRe Team from the University of Barcelona and Technical University of Catalonia.

Subtasks

Lifelog Semantic Access Task (LSAT)

Keywords

lifelogging, egocentric images, semantic image retrieval

1. INTRODUCTION

With the advances of wearable technologies during the last years, lifelogging has become a common trend nowadays. However, since lifelogging implies the collection of a huge amount of data, it requires powerful data management techniques to extract and retrieve the information of interest. The goal of the NTCIR Lifelog Semantic Access Task (LSAT) is to design new techniques for retrieving specific moments from the lifelog of a person. The challenge dataset consists of 90,586 images captured by three different users during a period of almost a month. Table 1 summarizes the characteristics of the dataset as well as of the users who contributed to gather the data. A subset of 89,593 images was enriched with tags of location and activity, and a 1,000 classes object classifier was used to assign object categories to the pictures.

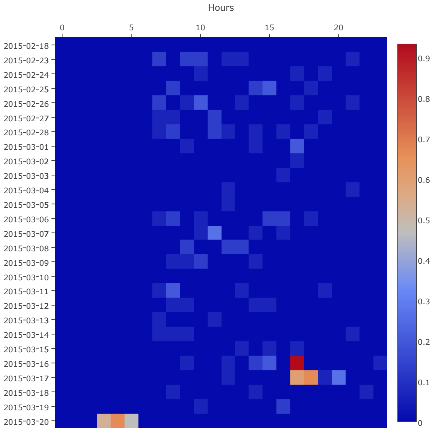


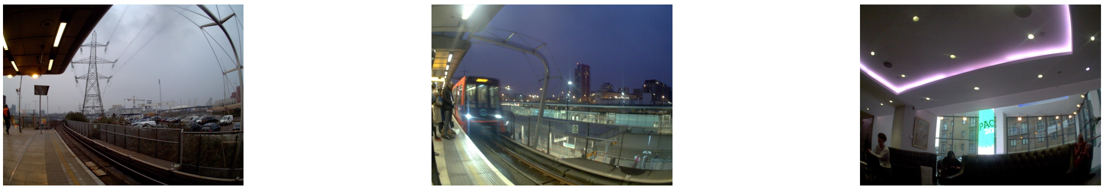
Figure 1: Heatmap of the sum of the histograms for the tags *train* and *train station* from the set of images of user *u1*.

In this paper, we describe LEMoRe, a new Lifelog Engine for Moments Retrieval, developed to participate in the NTCIR LSAT task. LEMoRe is based on the idea that lifelogging image annotations could provide the basic semantic context to retrieve a specific event. In other words, a particular moment of a person's life can be retrieved based on the objects present in that moment (food, laptop, beer, etc.), its location (bar, office, home, etc), the activity performed (shopping, working, etc.), and time along the day (morning, afternoon, etc), in an interactive fashion.

A first introduction to LEMoRe is given in sec. 2 and its architecture is detailed in sec. 3. In sec. 4, the required data preprocessing tasks are explained. A detailed list of functionalities offered by the engine is presented in sec. 5. A description of the performed run trials and their results is presented in sec. 6 and 7, respectively. Finally, sec. 8 provides some concluding remarks.

2. LEMORE

LEMoRe is an interactive Lifelog Engine for Moments Retrieval, whose front-end is a web-based user interface. Since LEMoRe relies on the semantic context (objects, places, ac-



(a) 2015-03-16 17:38:47

(b) 2015-03-17 18:32:15

(c) 2015-03-17 17:10:39

Figure 2: Top scores images for the three most probable dates and hours for the tags *train* and *train station.*

	User		
	u1	**u2**	**u3**
Gender	male	male	male
Weight (kg)	78	74	85
Height (cm)	173	183	176
Age	40	33	48
Number of days acquired	27	25	28
Number of tagged images	38,609	24,401	26,583

Table 1: Description of the dataset users.

	Number of tags
Activity	6
Clustered locations	35
Caffe network	1000
LSDA	3822
Merged tags	**4308**

Table 2: Number of tags per information source.

tivities and time) to retrieve a specific moment, the most important element considered in the interaction with the user is a heatmap of the semantic context in terms of tags scores by day and hour/minute, as shown in Fig. 1. Its purpose is to sort the images by the time, they were taken and highlight the ones that contain a certain input set of tags that express the context of a moment.

For example, if user *u1* is looking for a moment, when he was riding a red colored train, then the search could start by using the tags *train* and *train station.* The corresponding heatmap plot for this combination of tags (see Fig. 1) identifies three different dates, where the object *train* and the location *train station* might coincide in the images. After inspecting the images from the most probable hours of the suggested dates, the user could find the specific moment, he is looking for. In this case, some representative images that the actual search could retrieve are shown in Fig. 2, and for this example a good answer might be the image in Fig. 2b. Moreover, since the user is not aware of the many tags the system has, we implemented a tag suggestion service to help him at the beginning of his search.

Another key idea considered is that events may have similar settings at different times and dates. For instance, a moment like *watching the TV in the living room* is restricted to a place and can happen at the same time in different days or weeks. For the purpose of finding similar events across different days, LEMoRe can retrieve images with similar low-level features features, as it is further explained in sec. 3.

3. OVERALL ARCHITECTURE

As shown in Fig. 4, our proposed architecture for egocentric image retrieval consists of two main components: an image features engine, and a semantic engine. The image features engine retrieves images based on low-level features such as their color and textures characteristics, i.e. the task of this engine is to search for similar images. Specifically, our system uses auto color correlogram (CL), edge histogram (EH), joint composite descriptor (JCD), and pyramid histogram of oriented gradients (PHOG). The retrieval system is the open source Lucene Image Retrival engine (LIRE) [1].

The semantic engine retrieves images based on their tags (high-level concepts) and temporal information. The main idea of the semantic engine is that the tagged object detection, times, locations, and actions offer good semantic descriptors for events retrieval. The engine combines matrix numeric processing and database queries for object and temporal tags, respectively. Our engine offers four different operations as web components, that are fully described in sec. 5. These components were implemented in Python (NumPy) and (SQLite).

4. DATA PREPROCESSING

In order to achieve better results and faster response-times from our semantic engine, four data preprocessing tasks were carried out: 1) merging the semantic tags, 2) computing the score histogram for each tag per minutehour, 3) creating a tag similarity matrix, and 4) filling a database with this data. In particular, merging the semantic tags and the creation of a similarity matrix had the purpose of aggregating data, while the rest served for saving computing time during browsing the images.The following subsections describe the four data preprocessing tasks.

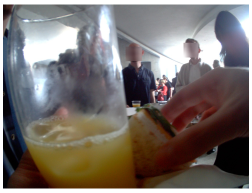
4.1 Tags Confidence

The tags confidence matrix contains the score values of all tags for each image in the dataset, where the rows and columns correspond to images and tags, respectively. The objective of this task is to merge in a single matrix the information about the location, activity and tags provided by the dataset and additionally the tags obtained from the LSDA object detector [2]. All the tags have a score value between 0 and 1 that represents the level of confidence of the annotation. Since the activity and location were manually annotated for some of the images, their scores were set to 1. In order to simplify the number of places, the locations provided were manually clustered. For instance, the words *Tesco* and *Lidl* were clustered together as *Supermarket.*

Since the tags scores provided by the original dataset were obtained using a CNN classifier trained on Caffe [3], only a small set of the images that present big centered objects might have been correctly classified. For instance, Fig. 3 shows the difference between a well-tagged image and badly-

Class ID	First class name	Score
673	mouse	0.17
527	desktop computer	0.13
508	computer keyboard	0.13
664	monitor	0.12
526	desk	0.12
782	screen	0.11

(a) Well-tagged image



Class ID	First class name	Score
876	tub	0.088
441	beer glass	0.044
435	bathtub	0.032
737	pop bottle	0.027
469	caldron	0.025
909	wok	0.024

(b) Badly-tagged image

Figure 3: Examples of the top five tags for a couple of images using the provided Caffe concepts by the NTCIR.

tagged image. Therefore, the LSDA object detector [2] was employed to enriching the number of concepts in the dataset. From its output, the three most probable detected objects were added as tags for each image with a value of 1.

The problem of merging tags from different sources is their repetition. So if a couple of tags had the same name, then its score for each image was calculated as $\max(s_1, s_2)$, where s_1 and s_2 are the original score values. Table 2 shows the final number of tags for the merged sources.

4.2 Tags Histogram

In this task, a score histogram h_t by day and minute/hour is created for each tag t. Moreover, each histogram is normalized by dividing it with its maximum value.

4.3 Tags Similarity

The tags similarity matrix represents a graph structure that links the level of similarity between tags. The similarity matrix is a symmetric matrix, whose values are between 0 and 1, where 0 and 1 are the lowest and highest level of similarity, respectively. The similarity between tags was obtained using WordNet [4] . The similarity value between two tags was calculated as the maximum similarity given by WordNet between the synonyms of both tags.

4.4 SQL database

In order to perform time-related queries, a SQL database was created. Its schema is focused on image ordered by its time, date, and user; Fig. 5 depicts the designed schema. In addition, the tags matrix row corresponding to each image was added as an attribute. This attribute is the link to perform operations with the image timestamps and the rest of the annotated data.

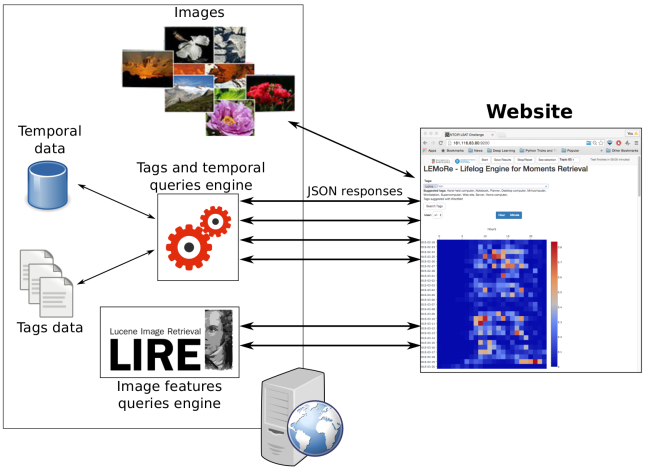

Images
Temporal data
Tags and temporal queries engine
JSON responses
Website
LEMoRe - Lifelog Engine for Moments Retrieval
Tags data
Lucene Image Retrieval
LIRE
Image features queries engine

Figure 4: Proposed image retrieval architecture.

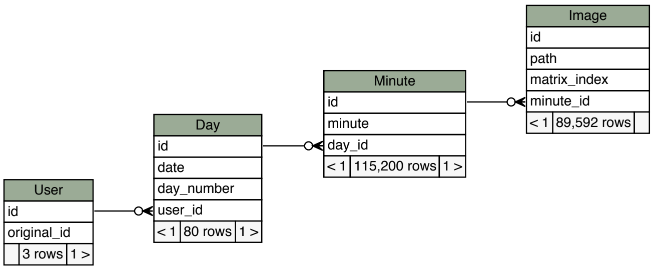


Figure 5: LEMoRe database schema.

5. FUNCTIONALITIES

The functionalities were designed to minimize the response time. Basically, their development followed two strategies: (1) using fast and efficient algorithms and (2) splitting the matrices of data in rows for fast memory access.

5.1 Scores histogram

The scores histogram functionality computes the sum of the tag scores histogram H by hour or minute given a set of n input tags. Essentially, it is an OR query operation that calculates:

$$H = \sum_{i=1}^{n} h_{t_i} \tag{1}$$

for a set of tags $\{t_1, t_2, \ldots, t_n\}$. This functionality sums the histograms already computed as stated in sec. 4.2.

5.2 Tag similarity

The tag similarity functionality suggests m semantically related tags given a set of n input tags. This service computes the set T of the m most similar tag indexes:

$$T = \underset{t_i \notin \{t_1,\ldots,t_n\}}{\arg\max} \sum_{i}^{n} s_{t_i}, \tag{2}$$

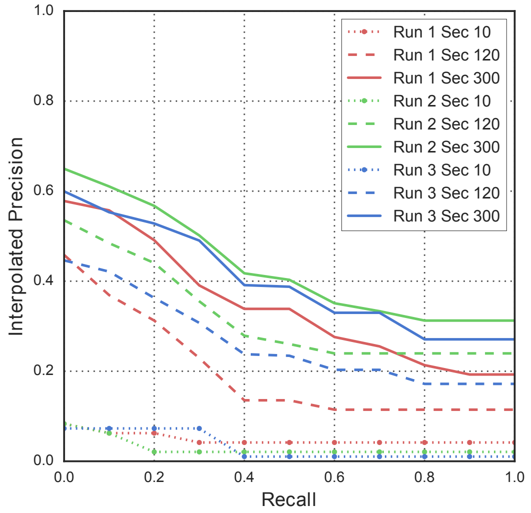


Figure 6: Event-level interpolated precision over recall for all submitted runs on seconds 10, 120, and 300.

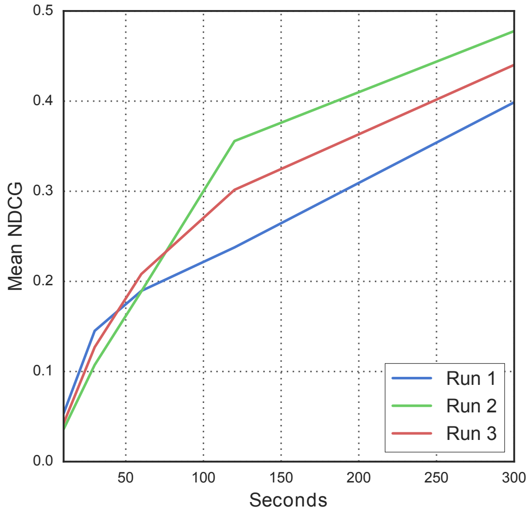


Figure 7: Mean normalized discounted cumulative gain (Mean NDCG) results over time for each run of event-level retrieval.

where t_i is an index of the similarity matrix and s_{t_i} one of its rows. In order to make this functionality faster, first it only loads in memory those requested rows from the similarity matrix. After the sum of the n rows is done, it returns the first m indexes using the Wirth partial sorting algorithm [5].

5.3 Images

The image functionality queries the database and returns a list of images given a user, date, and hour or minute. Additionally, it can sort the images by their sum class score given a set of n input tags. This is done by computing the class sum score:

$$T = \sum_{i=1}^{n} s_{t_i}, \tag{3}$$

where t_i is the tag index of each image, and then sorting them by it.

Algorithm 1: Image retrieval procedure for the challenge runs.

```
   Input: Moment description query
   Output: Event images
1  Read and specify user and time;
2  Think about relevant tags;
3  Look at "suggested tags";
4  Inspect the 10 most relevant images;
5  if any image is relevant then
6      recover the moment to inspect;
7  Inspect the heatmap: observe all red, orange, and
   light-grey moments;
8  if an answer is detected then
9      Search by:
       (I) Dates and times.
       (II) Image descriptors (CL, EH, JCD, PH).
       (III) Tags from retrieved images.
10 else
11     Think about alternative tags;
12 if the problem is "precision" then
13     goto end;
14 if the problem is "recall" then
15     goto Step 2 and explore the whole heatmap;
```

5.4 Top scored images

The top scored images functionality returns a list of m images with the highest class score given a set of n input tags. First, the functionality computes the minutes histogram sum of the requested n tags (sec. 5.1). Then, it obtains the largest $m/2$ minutes by partially sorting the resulting matrix H using [5]. Since it is expected that every minute has at least two images, then the sum of the tag scores is done for each image in every minute, thus resulting on m images. Finally, the sorted list of images by its scores is returned.

6. RUN TRIALS

In the Lifelog Semantic Access Task (LSAT) subtask, the participants have to retrieve a number of specific moments in a lifelogger's life. We define moments as semantic events or activities that happened throughout the day. We submitted three different runs to the challenge competition. Each run was performed by 4 different users, doing 12 out of the 48 queries each. Before querying the system, the users were instructed to follow the procedure described in Algorithm 1. The first run was made by users who were familiar to the competition. For the second run, a tag suggestion service was added in order to observe possible performance improvement of the system. This test was done by the same users as the test one, but performing a different set of queries each. The objective of the third run was to simulate a real-case scenario and test the usability of the system. Therefore, this run was made by users who were not familiar with the competition and also using the tag suggestion functionality.

7. RESULTS

On the tables 3 and 4, we present a quantitative overview for image- and event-level retrieval results for each run at

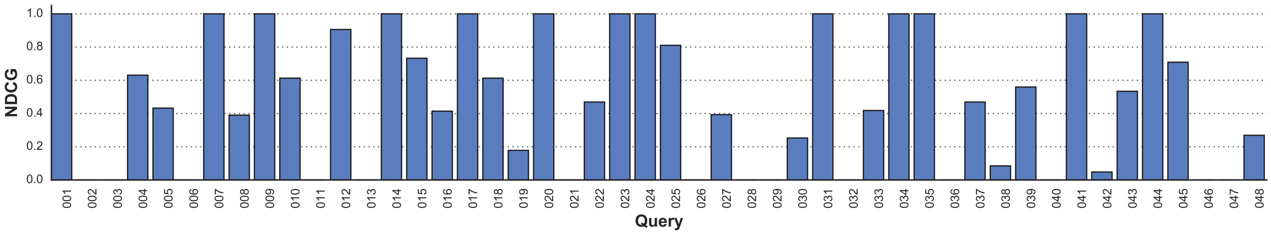


Figure 8: Normalized Discounted Cumulative Gain for event-level retrieval and for each query in second run after 300 seconds.

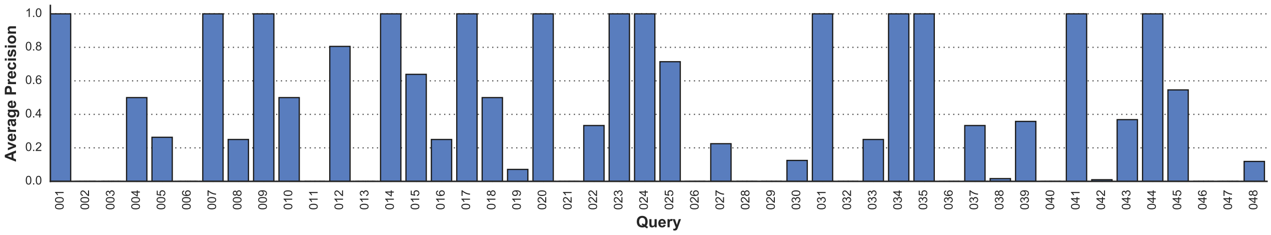


Figure 9: Average precision for event-level retrieval and for each query in second run after 300 seconds.

each evaluated time. Furthermore, the second run was the best according to the interpolated precision over recall curves for event-level queries shown in Fig. 6. It also points that our engine is well-balanced between precision and recall, and that more results are obtained after the first 10 seconds. Moreover, the mean normalized discounted cumulative gain (mean NDCG) for all event-level retrieval runs is depicted in Fig. 7; it shows that mean NDCG increases almost linearly over time for each run and the second run performed better. In addition, the normalized discounted cumulative gain (NDCG) and average precision (AP) values for each query are plotted in Fig. 8 and 9, respectively. Both values for each query show that the system performance is good, when the task is a single image moment or when the event takes place over a continuous period of time. On the other hand, the performance is not as good for moments that are sparse and not time contiguous.

8. CONCLUSIONS & FUTURE WORK

After a testing process of the LEMoRe in three submission runs, we obtained deeper insight about the system and its performance. Our results showed that its performance was better on moments that take place over a continuous time lapse rather than sparse. Moreover, we consider that better semantics improves event retrieval results, as the difference between the first and the second runs scores showed.

We will further work on improving the semantics of the system. Specifically, we explore techniques that produce tags for common daily settings that describe better the context. Additionally, we will also provide better interactive feedback to the user based on his/her selected images.

Since our final goal is to create a general purpose tool that is not limited to a specific kind of user, our efforts will be focused on two main objectives. First, the results showed a gap between experienced and non-experienced users that will be solved by making a friendlier interface. Second, the system still requires to further improve scalability and privacy issues.

9. ADDITIONAL AUTHORS

Additional authors: Maedeh Aghaei (University of Barcelona. email: maghaeigavari@ub.edu) and Marc Carné (Polythecnic University of Catalonia. email: marc.carne.herrera@estudiant.upc.edu).

10. REFERENCES

[1] Mathias Lux. Content based image retrieval with lire. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM '11, pages 735–738, New York, NY, USA, 2011. ACM.

[2] Judy Hoffman, Sergio Guadarrama, Eric Tzeng, Ronghang Hu, Jeff Donahue, Ross Girshick, Trevor Darrell, and Kate Saenko. LSDA: Large scale detection through adaptation. In *Neural Information Processing Systems (NIPS)*, 2014.

[3] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[4] C. Fellbaum. *WordNet: An Electronic Lexical Database*. Language, speech, and communication. MIT Press, 1998.

[5] N. Wirth. *Algorithms + Data Structures*. Prentice-Hall Series in Automatic Computation. Pearson Education Canada, 1976.

[6] Cathal Gurrin, Hideo Joho, Frank Hopfgartner, Liting Zhou, and Rami Albatal. Ntcir lifelog: The first test collection for lifelog research. In *Proceedings of the 39th International ACM SIGIR Conference on Research and*

	Run 1					Run 2					Run 3				
Seconds	**10**	**30**	**60**	**120**	**300**	**10**	**30**	**60**	**120**	**300**	**10**	**30**	**60**	**120**	**300**
Retrieved	16	159	317	528	1016	12	85	269	608	1153	11	73	166	349	752
Relevant	443	3978	5359	5755	6414	1417	2690	3076	5878	6606	406	1173	2793	4708	6306
Relevant-Retrieved	13	123	273	475	839	8	67	170	422	844	9	65	152	283	568
Mean Average Precision (MAP)	0.0048	0.0604	0.0965	0.1299	0.1834	0.0019	0.0211	0.0573	0.1372	0.2224	0.0119	0.0213	0.05	0.1075	0.1863
Average Precision. Geometric Mean	0.0	0.0001	0.0003	0.0009	0.0042	0.0	0.0001	0.0002	0.0019	0.0095	0.0	0.0001	0.0002	0.0007	0.0041
R-Precision	0.0048	0.0604	0.0973	0.1356	0.1939	0.0019	0.0213	0.0579	0.1386	0.2292	0.0149	0.0183	0.0476	0.1067	0.1882
Binary Preference (bpref)	0.0048	0.0604	0.0973	0.1356	0.1925	0.0019	0.0213	0.0579	0.1384	0.2277	0.0122	0.0183	0.0476	0.1059	0.1864

Table 3: Image-level summary statistics of all runs and seconds for the total number of images retrieved over all image queries.

	Run 1					Run 2					Run 3				
Seconds	**10**	**30**	**60**	**120**	**300**	**10**	**30**	**60**	**120**	**300**	**10**	**30**	**60**	**120**	**300**
Retrieved	6	18	29	46	104	11	30	49	79	140	5	13	28	67	127
Relevant	23	105	215	259	295	85	136	197	272	310	10	35	124	213	305
Relevant-Retrieved	4	13	21	34	73	8	20	32	56	94	4	12	22	48	82
Mean Average Precision (MAP)	0.0484	0.1215	0.1567	0.1895	0.3375	0.0281	0.0895	0.1626	0.3115	0.4204	0.0312	0.1054	0.1767	0.2562	0.3857
Average Precision. Geometric Mean	0.0	0.0002	0.0004	0.0013	0.0078	0.0	0.0001	0.0003	0.0036	0.0184	0.0	0.0001	0.0004	0.0016	0.0102
R-Precision	0.0484	0.1215	0.1466	0.1832	0.3159	0.0281	0.0902	0.1667	0.3052	0.4213	0.0208	0.095	0.1685	0.2586	0.3949
Binary Preference (bpref)	0.0484	0.1215	0.1466	0.1831	0.3365	0.0281	0.0901	0.1658	0.3041	0.4164	0.0208	0.095	0.1682	0.2496	0.3832

Table 4: Event-level summary statistics of all runs and seconds for the total number of images retrieved over all event queries.

Development in Information Retrieval, Pisa, Italy, July 2016. ACM.

[7] Cathal Gurrin, Hideo Joho, Frank Hopfgartner, Liting Zhou, and Rami Albatal. Overview of ntcir lifelog task. In *Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies*, NTCIR-12, National Center of Sciences, Tokyo, Japan, 7-10 June 2016. NTCIR-12.