Lifelog Moment Retrieval with Self-Attention based Joint Embedding Model

Hoang-Phuc Trang-Trung[1,3], Hoang-Anh Le[1,3], and Minh-Triet Tran[1,2,3]

[1] University of Science, VNU-HCM, Ho Chi Minh City, Vietnam
[2] John von Neumann Institute, Ho Chi Minh City, Vietnam
[3] Vietnam National University, Ho Chi Minh City,Vietnam
{tthphuc,lhanh}@selab.hcmus.edu.vn, tmtriet@fit.hcmus.edu.vn

Abstract. With the swift growth of technology, personal devices like cameras or healthcare sensors are more and more approachable, and many people use these devices to record their daily lives. So there is an increasing need for exploiting that enormous amount of data to understand more about how people live their lives. Thus, we introduce a novel interactive system to retrieve specific moments utilizing text-based queries. We propose Self-Attention based Joint Embedding Model (SAJEM) for that purpose. In our proposed method, we first extract visual and text features, then map them to a single common space, and calculate cosine distance for ranking. Besides, our system has two more auxiliary components using ResNet152 features and metadata of images to help users extend their query results. We also design a web application with an easy-to-use user interface to visualize and retrieve lifelog data. With this solution, we achieve the first rank in Lifelog Moment Retrieval task of ImageCLEF Lifelog 2020 with F1@10 score of 0.811.

Keywords: Lifelog retrieval · Image-Text cross-modal retrieval · User interface

1 Introduction

Lifelogging has been becoming more and more popular in the research communities. Lifelog dataset is the record of "lifeloggers" daily life, which mainly contains images captured by personal cameras and sensor data like location, heart rate, weight, audio, temperature, etc. With the rapid technological progress today, the amount of lifelog data becomes tremendous and almost impossible to handle manually. This motivates many researchers to develop a reliable and convenient system to exploit this lifelog data. The primary usage of this kind of system is to recall designated moments in the past, but it can also be used to analyze human social traits [1] or monitor user's health.

Copyright © 2020 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). CLEF 2020, 22-25 September 2020, Thessaloniki, Greece.

CEUR-WS.org/Vol-2696/paper_60.pdf

Many challenges and tasks in lifelogging have been proposed to create a competitive environment where people can address the problem and share knowledge about it. ImageCLEF [2] is one of those events which is held annually as part of the CLEF initiative labs. ImageCLEF Lifelog 2020 [3] includes 2 subtasks: Lifelog Moment Retrieval (LMRT) and Sport Performance Lifelog (SPLL). In this paper, we only focus on LMRT subtask. This year's lifelog dataset is enriched with more than 190,000 images in total (about 4.5 months of data from 3 lifeloggers, 1500-2000 images per day) along with visual concepts, semantic content, biometrics information, music listening history, and computer usage. Our mission is to find specific predefined moments described in 10 test topics which are handed out by organizers.

To resolve this problem, we propose an interactive retrieval system which allows user to express their queries in multiple ways. Our system depends on two major ideas:

- We aim to build a model that understand text queries instead of matching words. So we create a Self-Attention based Joint Embedding Model and train it on the COCO dataset [4] then use it on lifelog domain. We utilize a self-attention mechanism to learn the interaction between words in the text sentences and between objects in the images. With this model, users no longer need to overthink about choosing the right features to express a query, just input a sentence and see the results. We also add some auxiliary components like find similar semantic images or find images by metadata to make our system more powerful and reliable.
- An user-friendly web application allows users to split the query into multiple steps for better accuracy and easily choose the desired images for submission.

The rest of the paper is organized as follows: section 2 lists some related works to our research. In section 3, we introduce SAJEM model and describe our system in detail. Section 4 shows how we apply this system to LMRT subtask in ImageCLEF Lifelog 2020. Finally, section 5 concludes the paper.

2 Related Work

Lifelogging. Many lifelog retrieval systems have been developed in the past few years. Most of the previous works first extract visual concepts from images and find a way to save that data to a database for efficiently retrieving later on. They usually use textual tags to index an image. These tags include name of the detected objects [5–7], scene [5–7] or even optical characters [6] in the image. A different approach is used in [8] that they extract a combination of low-level features like HSV histogram and BRIEF features for example-based retrieval.

After the offline processing stage, a user interface is built to visualize the results and efficiently interact with the indexed database. For systems that leverage textual tags, they often use term queries combined with some techniques to improve accuracy: looking for synonym on thesaurus.com [5,7], utilizing pre-trained

word embedding [6] or BERT model [9] to obtain similar semantic words. Another interesting way of interaction is sketch-based retrieval which is enabled by video retrieval tools like VIRET [10], diveXplore [11] and vitrivr [12]. In [13], they even use virtual reality with distance-based and contact-based interaction to visualize and explore lifelog data.

Text-based image retrieval. Image-Text matching has been studied for a long time to support many essential applications: image captioning or cross-modal retrieval. Most of the existing approaches can be divided into two categories: designing a network to predict the similarity scores of image-text pairs or finding a joint embedding space under which we can compare image and text representations directly. The former idea is not suitable for retrieval tasks when the dataset is big because we need to run the network to calculate similarity scores between the query and all instances in the dataset at the online test stage. So we choose the latter idea to develop our model. SAJEM model is trained to minimize the Margin Ranking Loss between image and text embeddings with a negative sample mining strategy introduced in [14].

3 Interactive Lifelog Retrieval System

3.1 System Overview

After studying previous works on lifelog retrieval, we find out that there is a severe limitation in object detection based systems: they only focus on the existence of the objects, not interactions between them. We introduce Self-Attention based Joint Embedding Model (SAJEM), a novel model to leverage the interaction between objects and even the context of images. We integrate the model with two more auxiliary components to form a novel system which can handle multiple types of requests:

- **Query by text sentence:** The best way to express a query is through a free text sentence. Our system takes a sentence as the input, extract its feature, and map to joint embedding space. Then we use that sentence embedding to match with all image embeddings to find the most relevant images to that sentence.
- **Query similar images:** We calculate the cosine distance between the feature of a query image with all pre-extracted features of lifelog data and output images with the smallest distance. Features are extracted from ResNet152 model to capture the semantic meaning of images.
- **Query by metadata:** Filter data by places and time metadata provided by the organizers.

Figure 1 visualizes components and data flow in our system. At the offline stage, SAJEM and ResNet features are extracted for all images in lifelog dataset. Furthermore, we separate the web application into frontend and backend: the frontend module provides an user interface to help users interact with the system, and the backend handles requests from frontend and contacts with models and database. We briefly explain each component of the system in the following subsections.

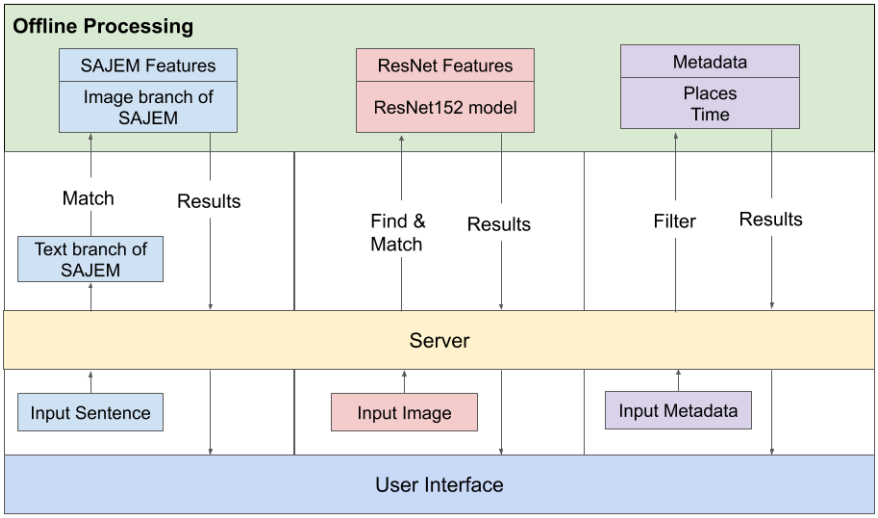


Fig. 1: Main components of the proposed system. There are 3 components corresponding to 3 types of input: free text sentence, image and metadata. Features of all images in the lifelog dataset are pre-extracted at offline processing stage to reduce computation time at online retrieval stage

3.2 Self-Attention based Joint Embedding Model

The overall architecture of our proposed model is shown in Figure 2. Our model consists of two branches corresponding to two domains we are working on: image and text domains. In the image branch, we begin with the features of detected regions in the image generated by the Bottom-Up Attention model [15]. We then use a Self-Attention Module to learn the interaction between image regions and build a single vector representation for the whole image. In the text branch, we use RoBERTa model [16] to learn the representation for an input sentence. Finally, Image/Text Feature Encoders are used to project corresponding features to the joint embedding space. Both branches of the model take advantage of the self-attention mechanism (Self-Attention Module in the image branch and RoBERTa in the text branch). Thus we call our model as Self-Attention based Joint Embedding Model.

Bottom-Up Attention Faster R-CNN: First, we extract the object-level features of an image. Faster R-CNN [17] is a state-of-the-art model in object detection. It is a two-stage detector. In the first stage, a Region Proposal Network is used to predict multiple bounding box proposals of different scales and aspect ratios. Then they use non-maximum suppression to reduce the number of proposals. In the second stage, each box proposal is transformed into a small feature map by Region of Interest pooling layer then feeds into a CNN to predict class label and class-specific bounding box refinements.

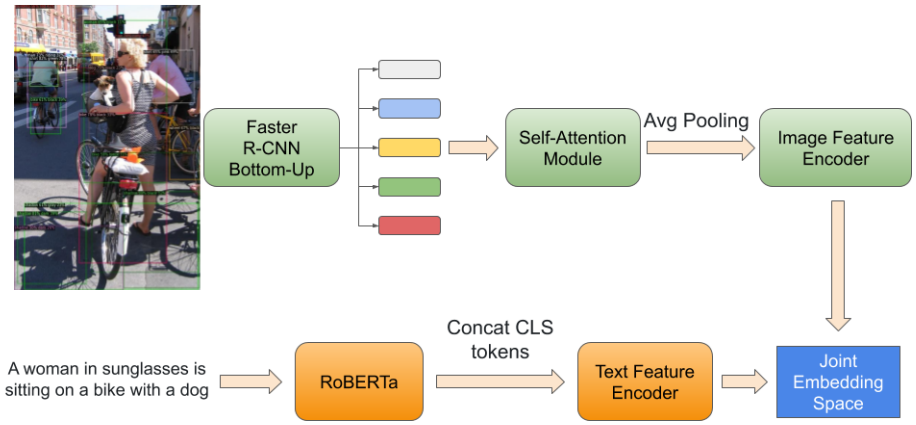


Fig. 2: Overall architecture of SAJEM. Image branch (top) detects regions in the image and then learn the interactions between these regions using Self-Attention Module to represent for that image. Text branch (bottom) uses RoBERTa model to encode the text sentence.

In Bottom-Up and Top-Down attention paper [15], they use Faster R-CNN with ResNet101 backbone and introduce a simple "hard" attention mechanism: only selects few regions with highest class detection probabilities. The feature of each region is defined as the mean-pooled convolutional feature of that region, so the dimension of each feature vector is 2048. To pretrain this bottom-up model, they initialize Faster R-CNN with ResNet101 (pretrained from ImageNet) then train on the Visual Genome dataset. To learn good feature representations, they add an additional training output for predicting attribute classes along with the old object classes.

We use the Bottom-Up pre-trained model available on GitHub[4]. Each image is represented by a set of feature vectors (each vector corresponding to a detected region in the image) $I = \{r_1, r_2, ..., r_k\}, r_i \in R^D$ (in this case $D = 2048$). Due to hardware limitations, we only select 15 regions with the highest probabilities.

Self-Attention Module: We adopt the idea of Multi-Head Self-Attention from Transformer model [18] with some refinements: use Dot-Product Attention instead of Scaled Dot-Product Attention and only use one head when applying attention.

As we mention above, each image is now represented by a set of vectors $I = \{r_1, r_2, ..., r_k\}, r_i \in R^D$. We can view it as a matrix $I \in R^{k \times D}$. Then, we

[4] https://github.com/airsplay/py-bottom-up-attention

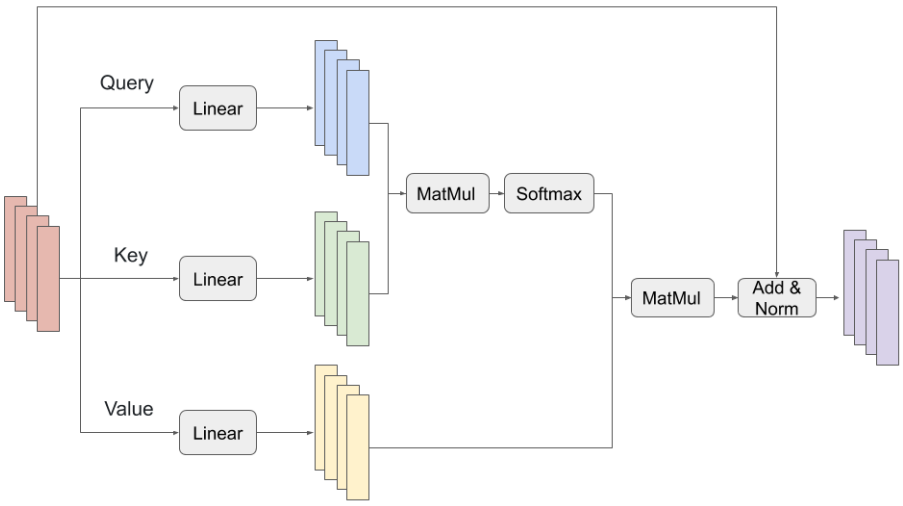


Fig. 3: Self-Attention Module. This module takes image region features from Bottom-Up Faster-RCNN as input and apply Dot-Product Attention on these features.

apply Self-Attention module as follows:

$$Q = IW_Q + b_Q$$

$$K = IW_K + b_K$$

$$V = IW_V + b_V$$

$$I^* = Softmax(QK^T)V$$

where $W_Q, W_K, W_V \in R^{D \times D}$ are the weight matrices and $b_Q, b_K, b_V \in R^D$ are biases of linear transformations, respectively.

After applying Self-Attention, we add the residual connection to the original image features, then use Layer Normalization:

$$Output = LayerNorm(I + I^*)$$

Finally, we apply Average Pooling over all regions of the image to achieve one D-dimension vector representation for each image.

RoBERTa as a text feature extractor: In recent years, Transformer-like models [18] have become very popular in the NLP field. These models leverage attention mechanism to learn good text representations through Mask Language Model and Next Sentence Prediction tasks, then apply it to many down-stream tasks and achieve state-of-the-art results. In particular, BERT (Bidirectional Encoder Representations from Transformers) [19] use bidirectional transformer

(both left-to-right and right-to-left direction) to produce context-aware representations.

In this work, we use RoBERTa model [16], which is the BERT model with some training tricks and more data. We use the pretrained RoBERTa-base model provided by HuggingFace [20] available on GitHub[5]. Each token corresponds to each word in text sentence or special character like 'CLS' for classification purpose, 'SEP' for separating sentences, etc. Each token is represented by a 768-dim vector. We concatenate the last 4 'CLS' tokens of the last four layers of the model to create a 3072-dim vector to represent for the text sentence. We also fine-tuned this pre-trained model while training to adapt it to this specific domain.

Joint Embeddings Learning: Given an image and its corresponding caption, we extract their feature vectors by the models mentioned above. Then we project these vectors into a common space by feeding them into Image or Text Feature Encoder, respectively. These encoders are just neural networks with one hidden layer. After transformation, we achieve same dimension vectors for both image and text caption: $\phi(I)$ and $\phi(C)$.

We adopt the Margin Ranking Loss from [14], which penalizes the model according to negative samples.

$$L(\phi(I), \phi(C)) = max(0, \alpha + S(\phi(I), \phi(C)^-) - S(\phi(I), \phi(C)))$$

$$+max(0, \alpha + S(\phi(I)^-, \phi(C)) - S(\phi(I), \phi(C)))$$

where α is non-negative margin constant. $\phi(I)^-$ and $\phi(C)^-$ are hardest negatives in the current mini-batch for $\phi(C)$ and $\phi(I)$, respectively. S is the similarity function.

Training Process: We train our model on MS COCO dataset [4]. Each image in the training set has five captions, so we can create five training samples with one image. The model was implemented using PyTorch framework and trained on NVIDIA Tesla P100 GPU provided by Google Colab. We also apply some tricks in the training process to improve the result: use different learning rates for RoBERTa and other components of the model, use Adam optimizer and linear learning rate scheduler, freeze RoBERTa model in the first epoch to warm up other components. Finally, we evaluate our model on the MS COCO 5k test set and achieve Recall@10 of 0.732.

3.3 Query similar images using ResNet152 features

Many experiments have shown that feature in the layer before the classification layer of a Convolutional Neural Network can be efficient for representing an image. Adopting that idea, we feed all lifelog images into a ResNet152 model pre-trained on ImageNet to achieve a 2048-dimension vector for each image and

[5] https://github.com/huggingface/transformers

then index that the feature vector by image name to retrieve later on easily.

At the online stage, given an image, we can find the most similar images in terms of semantic meaning by just comparing the cosine distance between features of a reference image and all images in the lifelog dataset. We only allow users to input image in the lifelog dataset, not an arbitrary image so we can easily get the reference feature by searching in the feature dataset, not worry about loading ResNet152 model to memory and computation overhead at runtime.

3.4 Query by metadata

The organizers provide us various information collected by sensors for each moment in the lifelog dataset: date and time, GPS coordinates, semantic name for locations, elevation, speed, heart rate, etc. In Lifelog Moment Retrieval task, we figure out that time and location are essentially useful for many types of queries like "Find the moments when lifelogger was eating seafood in a restaurant in the evening time". To make use of such information, we integrate some features to our system:

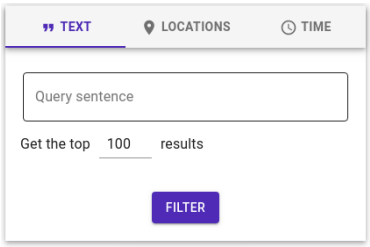
- Get all images in a specific time range of a day, e.g., from 5:30 PM to 9:00 PM.
- Get all images taken in a specific location, .e.g, Home, Dublin City University (DCU).
- Get all images taken in a time interval before going to a location, e.g., 40 minutes before going Home.
- View timeline: using time metadata to traverse back and forth in time from an image.

3.5 User interface

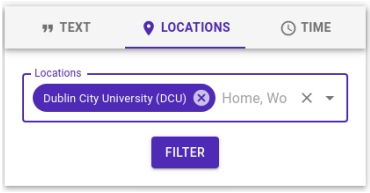
After trying out our model with many queries, we find out that lengthy sentences can cause the model to pay attention to too much information and lead to worse performance. To tackle this problem, we build a user interface that allows users to split a query into multiple steps, each step responsible for a small chunk of information of that query.

Result of one query can contains a long list of images so we use pagination to prevent users from being overwhelmed with hundreds of images. Moreover, when using pagination, the browser can render a small amount of images at a time instead of loading all at once, thus reducing the waiting time for better user experience.

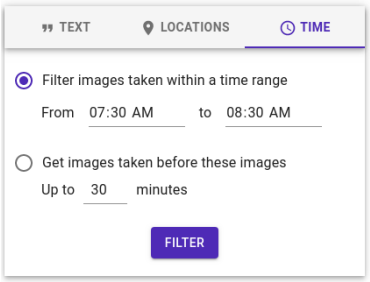
We also support users to choose desired images and output submission file for each query. Furthermore, the result page has drag-and-drop feature to rearrange the order of images and remove button to get rid of unwanted images.



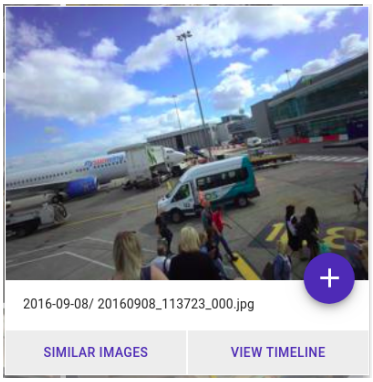
(a) Query by text sentence



(b) Query by location metadata



(c) Query by time metadata



(d) Query by ResNet152 feature and View timeline

Fig. 4: User interface for each component of our retrieval system

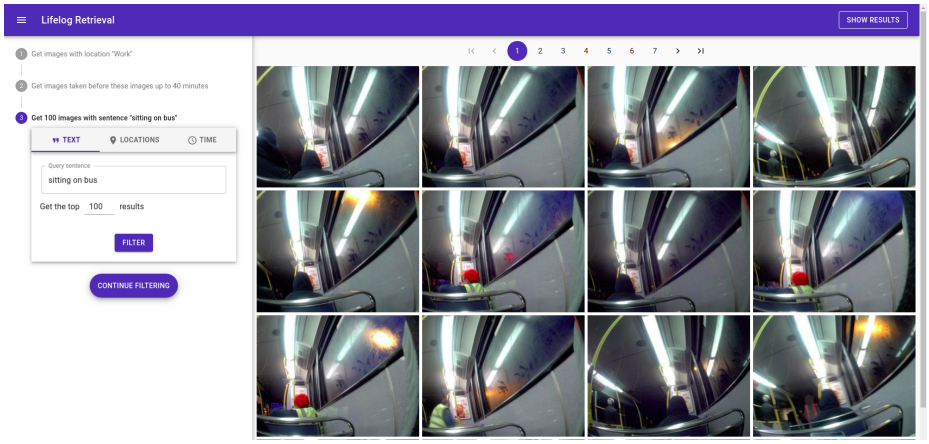


Fig. 5: Example of multi-step query and pagination to visualize result

4 Result

4.1 Result in ImageCLEF Lifelog 2020

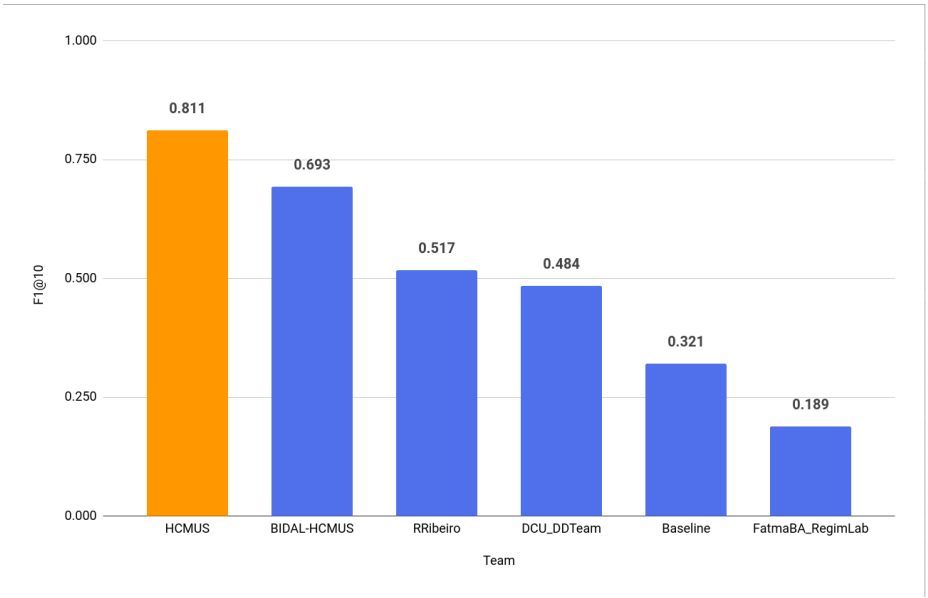


Fig. 6: Leaderboard of ImageCLEF Lifelog 2020 LMRT subtask

Our team use the name "HCMUS" to participate in ImageCLEF Lifelog 2020 - Lifelog Moment Retrieval subtask. Figure 6 show that our team achieve the highest result in term of F1-score at 10 compared to others .

Table 1 shows our detailed result of our best run with each test topic which contains Precision at 10 (P@10), Cluster Recall at 10 (CR@10) and F1 score at 10 (F1@10). Almost all the test topics have high CR@10. This means our system manages to find all the moments that are relevant to the query. Our system is able to do this because of the capability to split a query into multiple steps and handle them one by one. Moreover, thanks to the flexibility of language, we can express a text query in many different ways. For example, we can change the query *"He is repairing his car with a wrench in his hand"* to passive voice *"His car is being repaired with a wrench"* or using synonym *"He is using a spanner to repair his car"*. Like mentioned before, we can also split this query into multiple steps: *"He is repairing his car"* and continue querying on the returned results *"He is holding a wrench"*. This can open a huge potential for our system and help it reduce the risk of handling too long sentence which can hurt the performance.

Topic	P@10	CR@10	F1@10
1. Praying Rite	1.00	1.00	1.00
2. Lifelog data on touchscreen on the wall	1.00	0.75	0.86
3. Bus to work - Bus to home	1.00	1.00	1.00
4. Bus at the airport	0.60	0.50	0.55
5. Medicine cabinet	0.90	0.78	0.83
6. Order Food in the Airport	0.70	0.67	0.68
7. Seafood at Restaurant	1.00	0.40	0.57
8. Meeting with people	0.50	1.00	0.67
9. Eating Pizza	0.90	1.00	0.95
10. Socializing	1.00	1.00	1.00
Average	0.86	0.81	0.81

Table 1: Detailed result of our best run with each test topic in ImageCLEF 2020 LMRT subtask

4.2 Experiment with LMRT test topics

In this section, we show our system's results for some test topics and the process to achieve that results.

TOPIC "MEDICINE CABINET"

Description: Find the moment when u1 was looking inside the medicine cabinet in the bathroom at home.

Narrative: To be considered relevant, u1 must be at home, looking inside the opening medicine cabinet beside a mirror in the bathroom. The moments that u1 was looking inside the medicine cabinet in other places (not at home and not in the bathroom) or u1 was looking at the closed medicine cabinet are not considered to be relevant.

Our solution: We can find relevant images for this topic by inputting one of these sentences *"looking inside the medicine cabinet in the bathroom"* or *"looking for medicine in the bathroom"*. Moreover, we observe that images with "opened medicine cabinet" are very similar to each other, so we use "Find similar images by ResNet152" feature to find remaining images.

Fig. 7: Top 4 images when querying with text sentence *"looking inside the medicine cabinet in the bathroom"*. The first and the fourth image can be considered relevant.

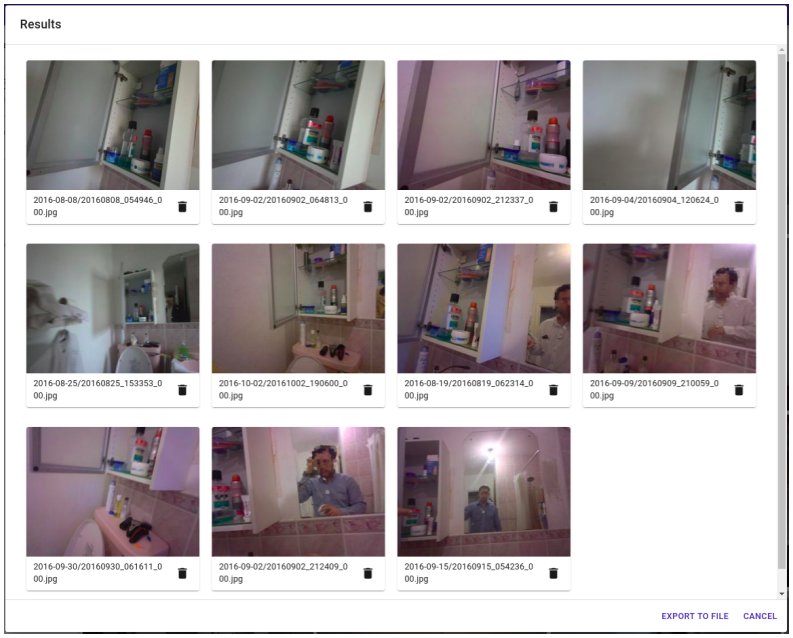


Fig. 8: Result for topic "Medicine cabinet"

Although we can not find all relevant moments for this topic, we manage to find some "hard" images. For example, in the fourth image in figure 8, the cabinet is on the edge of the image which makes it really hard to be detected.

TOPIC "SOCIALIZING"

Description: Find the moments when u1 was talking to a lady in a red top, standing directly in front of a poster hanging on a wall.

Narrative: To be relevant, the u1 must be talking with a woman in red, who was standing right in front of a scientific research poster.

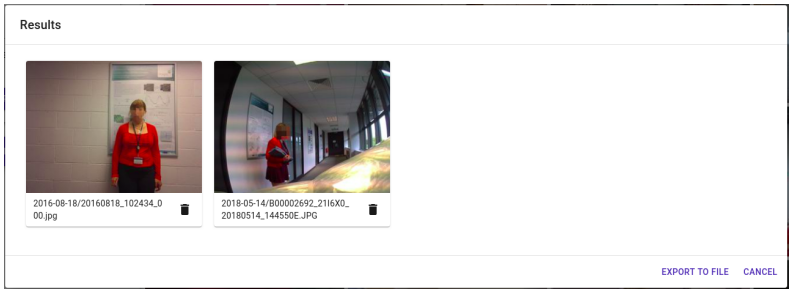


Fig. 9: Result for topic "Socializing"

Our solution: The ground truth of this topic contains two images. The first one we can easily retrieve with sentence *"a woman in red top standing in front of a poster"*. We try dividing the query into two steps: first, get 1000 images with sentence *"a woman in red"* then continue to filter with sentence *"poster on the wall"* and search through the result to find the second image successfully.

TOPIC "BUS TO WORK - BUS TO HOME"

Description: Find the moment when u1 was getting a bus to his office at Dublin City University or was going home by bus.

Narrative: To be relevant, u1 was on the bus, and the destination is his home or his workplace. The moments that u1 was waiting at the bus stop or u1 was traveling on any other public transportations, or the destination is not his home/workplace are not considered relevant.

Our solution: We use the "Query by metadata" feature to filter all images which had been taken up to 40 minutes before the lifelogger arrived at Work, Dublin City University or Home. Then, we continue filtering on these images with sentence *"sitting on bus"* to get relevant images. Finally, we use the "View timeline" feature to choose the best suitable candidates for submission.

Figure 5 shows the results when querying *"sitting on bus"* on the images which had been shot up to 40 minutes before the lifelogger got to Work.

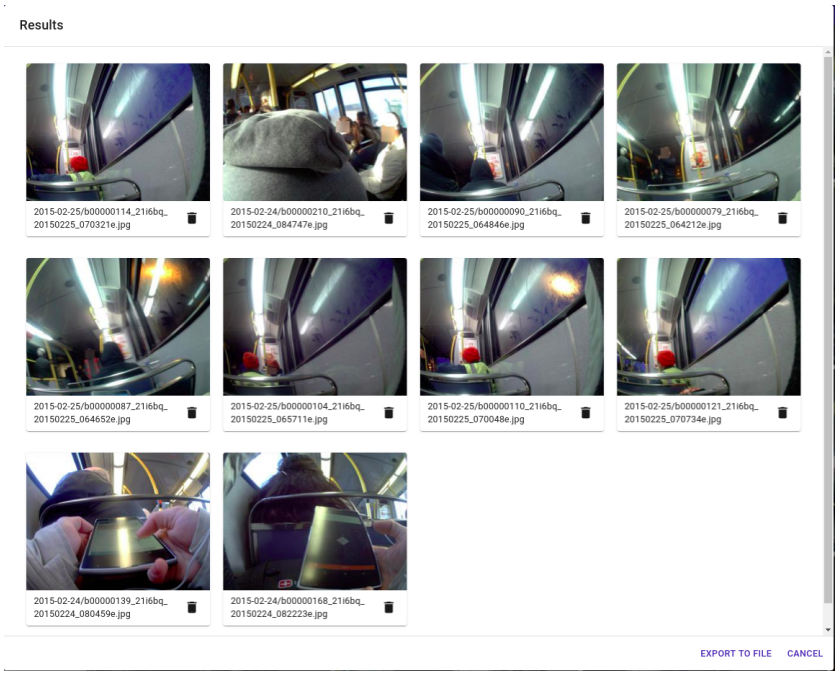


Fig. 10: Result for topic "Bus to work - Bus to home"

5 Conclusion

In this paper, we propose a novel lifelog retrieval system which mainly focuses on free text query relied on Self-Attention based Joint Embedding Model. We also integrate two more components: query by ResNet152 and query by metadata to make the system more robust and reliable. We create a web application with a user-friendly user interface to help users interact with these models and visualize lifelog data. We use this system to participate in ImageCLEF Lifelog 2020 LMRT task and achieve the first rank with F1@10 at 0.811.

Although our system performs well on this task, it still has some typical drawbacks, as in many deep learning models: lack of explanation and reliability for the result. Therefore, the model can output irrelevant images in some cases.

Acknowledgement

This research is supported by Vingroup Innovation Foundation (VINIF) in project code VINIF.2019.DA19.

References

1. T. Dinh, H. Nguyen, and M.-T. Tran, "Social relation trait discovery from visual lifelog data with facial multi-attribute framework," pp. 665–674, 01 2018.
2. B. Ionescu, H. Müller, R. Péteri, A. B. Abacha, V. Datla, S. A. Hasan, D. Demner-Fushman, S. Kozlovski, V. Liauchuk, Y. D. Cid, V. Kovalev, O. Pelka, C. M. Friedrich, A. G. S. de Herrera, V.-T. Ninh, T.-K. Le, L. Zhou, L. Piras, M. Riegler, P. l Halvorsen, M.-T. Tran, M. Lux, C. Gurrin, D.-T. Dang-Nguyen, J. Chamberlain, A. Clark, A. Campello, D. Fichou, R. Berari, P. Brie, M. Dogariu, L. D. Ştefan, and M. G. Constantin, "Overview of the ImageCLEF 2020: Multimedia retrieval in lifelogging, medical, nature, and internet applications," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, vol. 12260 of *Proceedings of the 11th International Conference of the CLEF Association (CLEF 2020)*, (Thessaloniki, Greece), LNCS Lecture Notes in Computer Science, Springer, September 22-25 2020.
3. V.-T. Ninh, T.-K. Le, L. Zhou, L. Piras, M. Riegler, P. l Halvorsen, M.-T. Tran, M. Lux, C. Gurrin, and D.-T. Dang-Nguyen, "Overview of ImageCLEF Lifelog 2020:Lifelog Moment Retrieval and Sport Performance Lifelog," in *CLEF2020 Working Notes*, CEUR Workshop Proceedings, (Thessaloniki, Greece), CEUR-WS.org <http://ceur-ws.org>, September 22-25 2020.
4. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.
5. N.-K. Le, D.-H. Nguyen, T.-H. Hoang, T.-A. Nguyen, T.-D. Truong, D.-T. Dinh, Q.-A. Luong, V.-K. Vo-Ho, V.-T. Nguyen, and M.-T. Tran, "Smart lifelog retrieval system with habit-based concepts and moment visualization," in *Proceedings of the ACM Workshop on Lifelog Search Challenge*, pp. 1–6, 2019.
6. C.-C. Chang, M.-H. Fu, H.-H. Huang, and H.-H. Chen, "An interactive approach to integrating external textual knowledge for multimodal lifelog retrieval," in *Proceedings of the ACM Workshop on Lifelog Search Challenge*, pp. 41–44, 2019.

7. T.-K. Le, V.-T. Ninh, D.-T. Dang-Nguyen, M.-T. Tran, L. Zhou, P. Redondo, S. Smyth, and C. Gurrin, "Lifeseeker: Interactive lifelog search engine at lsc 2019," in *Proceedings of the ACM Workshop on Lifelog Search Challenge*, pp. 37–40, 2019.
8. I. Nguyen Van Khan, P. Shrestha, M. Zhang, Y. Liu, and S. Ma, "A two-level lifelog search engine at the lsc 2019," in *Proceedings of the ACM Workshop on Lifelog Search Challenge*, pp. 19–23, 2019.
9. S. P. Nguyen, D. H. Le, U. H. Pham, M. Crane, G. Healy, and C. Gurrin, "Vielens, an interactive search engine for lsc2019," in *Proceedings of the ACM Workshop on Lifelog Search Challenge*, pp. 33–35, 2019.
10. J. Lokoč, T. Souček, P. Čech, and G. Kovalčík, "Enhanced viret tool for lifelog data," in *Proceedings of the ACM Workshop on Lifelog Search Challenge*, pp. 25–26, 2019.
11. B. Münzer, A. Leibetseder, S. Kletz, M. J. Primus, and K. Schoeffmann, "lifexplore at the lifelog search challenge 2018," in *Proceedings of the 2018 ACM Workshop on The Lifelog Search Challenge*, pp. 3–8, 2018.
12. L. Rossetto, R. Gasser, S. Heller, M. Amiri Parian, and H. Schuldt, "Retrieval of structured and unstructured data with vitrivr," in *Proceedings of the ACM Workshop on Lifelog Search Challenge*, pp. 27–31, 2019.
13. A. Duane, C. Gurrin, and W. Huerst, "Virtual reality lifelog explorer: lifelog search challenge at acm icmr 2018," in *Proceedings of the 2018 ACM Workshop on The Lifelog Search Challenge*, pp. 20–23, 2018.
14. F. Faghri, D. J. Fleet, J. R. Kiros, and S. Fidler, "Vse++: Improving visual-semantic embeddings with hard negatives," *arXiv preprint arXiv:1707.05612*, 2017.
15. P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6077–6086, 2018.
16. Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.
17. S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.
18. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.
19. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
20. T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "Huggingface's transformers: State-of-the-art natural language processing," *ArXiv*, vol. abs/1910.03771, 2019.