Natural Language Access to Video Databases

Danny Francis, Paul Pidou, Bernard Merialdo, Benoit Huet
Data Science Department, EURECOM
Sophia Antipolis, France
{Firstname.Lastname}@eurecom.fr

***Abstract*—This paper deals with natural language access to video databases. Two approaches are proposed: in the first one we use queries to find images similar to video keyframes, and in the second one we generate text descriptions from keyframes and compare them with queries. We propose four implementations of these approaches: one implementation of the first approach, two implementations of the second one and one implementation mixing both approaches. The results of our implementations are discussed, in particular regarding the visual content of natural language queries.**

I. INTRODUCTION

Pattern Recognition techniques have recently incurred a breakthrough in performance, especially in Natural Language and Computer Vision. This opens the way to new applications in the management of large amounts of multimedia data. One interesting application is the possibility to use Natural Language to easily access the content of large video databases. A huge amount of video material is recorded and stored every day, generally with a limited description of the content. This creates a need for techniques that will allow users to easily specify a description in Natural Language and automatically retrieve interesting video segments.

In Natural Language Processing, Recurrent Neural Networks (RNN) [7] have become the most efficient form of language modeling. Word embeddings [6][12], which assign to each word a vector of scalar values, have shown to be very effective in representing the semantics of Natural Language. In Computer Vision, Convolutional Neural Network allowed to build very performing concept detectors [11][3]. When combined with language models, these networks allowed to produce a text description of any picture [5], or describe the various objects appearing in a scene [9]. Such combinations of techniques allow now to perform video search without any query example[10][14], using a simple text description.

TRECVID [1] is an international evaluation campaign organized by the National Institute of Standards and Technology (NIST) aimed at comparing techniques for the retrieval of Digital Video. In 2016, one of the proposed task was Ad-hoc Video Search (AVS). This is a new task where the goal is to retrieve the video shots in a large database that match a short textual topic description. The test database comes from the Internet Archive, and contains 600 hours of video, representing about 300,000 shots. 30 test topics are provided by NIST to the participants. Each participant can submit up to four runs, each run being a ranked list of at most 1,000 shots for each of the 30 test topics. Evaluation is performed manually by NIST annotators and measured using the Mean Inferred Average Precision (which is a statistical approximation of the Mean Average Precision). In the remainder of this paper, we will say for convenience "Mean Average Precision" instead of "Mean Inferred Average Precision".

In this paper, we present the main approaches that we have considered to construct our four runs to the AVS task. In particular, we wanted to explore two orthogonal strategies:

- from the text topic, interrogate web image search engines to collect examples of relevant pictures, then use these pictures to build a visual model, which in turn will select the best keyframes in the test database.
- from the test keyframes, automatically generate a text description, and then match this text description with the topic.

In order to implement these strategies, we used the following tools and services, which are freely available from the internet:

- to get example images for a topic, we used the Google ImageSearch engine [2]. This search engines allows to enter a text query and returns a list of corresponding images. The exact mechanism to retrieve those images is not published, however it is likely to be largely based on the textual context of the pages where these images appear. Although a number of other image search services are available, we limited ourselves to this only one by lack of time. For each topic, we kept only the first 100 images returned, as more and more irrelevant images occur when we go deeper in the result list.
- to get a text description from an image, we used several tools:
  - the VGG Deep Networks [3], which have been trained on part of the ImageNet database and can analyze an image to provide scores for 1,000 predefined concepts,
  - the ImageNet Shuffle [4], which provides classifiers trained on a larger share of the ImageNet database, and analyze images to produce scores for up to 13,000 concepts,
  - the NeuralTalk [5] package, which generates sentences describing the visual content of images.
- to compare visual contents, we compute a visual feature vector for an image by applying the VGG Deep Network

to each image and extracting the outputs of the one-before-last and two-before-last layers, to build visual vectors. The similarity between visual vectors is computed as the usual scalar product, sometimes with normalization.

- to compare textual content, we use the GloVe vector representations of words [6], to build a textual vector from either the topic description, the concept name or the descriptive sentence. The similarity between textual vectors is again computed as the usual scalar product.

Many combinations of these modules are possible, as well as different values of the parameters involved. In order to choose the combinations to be used in the final runs, we performed a number of experiments on the development collection. We ran several systems using the 48 development topics, and applied them on the development videos. Then, we manually annotated the 10 best keyframes returned for each system and each topic. This gave us some indications of which system would have the greater performance. We observed that the performance of very different approaches varied greatly depending on the topic, so in the final runs, we also chose to provide a selection of the different combinations that we tried.

II. Description of the Runs

A. Generic Architecture

The following figure illustrate the generic architecture that we have put in place, corresponding modules. The green modules represent text-based information, the blue modules contain visual information, the yellow modules represent similarity computations. We tried various combinations to define the four runs that we submitted to the final evaluation.

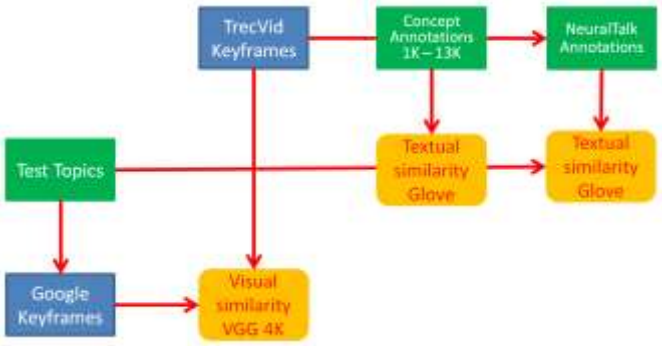

Figure 1 - Description of our runs

All our runs are of the "Fully Automatic" category, since no manual processing was done at any stage, and with the "D" training type, as we are using tools which were trained on data external to TRECVID.

B. RUN 1 "GoogleSearch + VGG 4K"

For each of the topic, we performed a search using the Google Image engine, and retained the first 100 pictures of the ranked list. To each image, we applied the VGG Deep network, and kept the one-before-last layer as feature vector of dimension 4K. We applied the same visual processing to each of the TRECVID keyframes in the test collection, and ranked them according to a Nearest Neighbor distance from the Google images.

C. RUN 2 "ImageShuffle + GloVe300"

We used the ImageShuffle system to obtain scores for 13,000 concepts, which we used as feature vectors for each TRECVID keyframe. We used these scores as weights to compute a semantic vector of dimension 300 by a linear combination of the 13,000 GloVe vectors corresponding to the concepts. For each topic, we constructed a semantic vector of dimension 300 by averaging the GloVe vectors of the words appearing in the topic. Then we used the cosine similarity to find the images whose semantic vectors were most similar to the topics.

D. RUN 3 "NeuralTalk + GloVe300"

We used the NeuralTalk system to generate text descriptions for each of the TRECVID keyframes. Then, we built a semantic vector of dimension 300 by averaging the GloVe vectors of the words appearing in these descriptions. We did the same for the test topics. Finally, we used again the cosine similarity to find the images whose semantic vectors were most similar to the topics.

E. RUN 4 "Global Average"

During the development phase, we experimented with a number of combinations of the modules that we have described, using different dimensions, different projections, different layers, different similarity measures. We evaluated these combinations with a minimal annotation on the development collections, by pooling the 10 best pictures for each of the training topics. This gave us an indication of which combinations could be the most efficient, and helped us in the selection of the combinations for the final runs to be submitted. As we noticed that different combinations had very different performances of different topics, we tried to get the best of all combinations by averaging the results of 32 combinations that we had found to be of reasonable performance. As the similarity scores are not always comparable between different combinations, we introduced for each combination an artificial score computed as the inverse rank of each image in the result list. The average of these 32 inverse ranks is the final score for this run.

III. Evaluations

The result (Mean Average Precision, or MAP) obtained by our four runs are the following:

TEAM	RUN	MAP
EURECOM	2	0.024
EURECOM	1	0.011
EURECOM	4	0.01
EURECOM	3	0.002

The following graph shows how they are located within the full set of (Fully Automatic) submissions from all participants (circles correspond to EURECOM submissions):

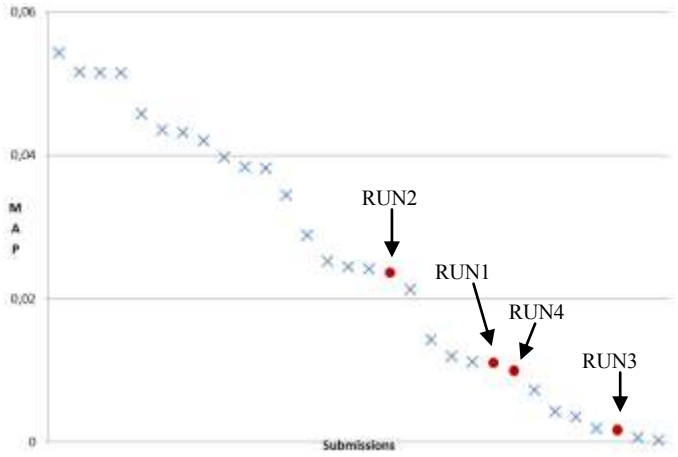

Figure 2 - Results of the AVS task

We can observe that our best run is RUN2, which is based on the ImageShuffle system, and has obtained a performance quite similar to the MediaMill team (which has developed ImageShuffle). The runs using Google Search or the full average have surprisingly very similar performance. RUN3, based on NeuralTalk, performed quite poorly, probably because of the mismatch between the test topics and the type of annotations on which NeuralTalk was trained.

IV. Analysis of the Results

We noticed that our models did not perform equally on all topics. In particular, RUN1 performed better on some topics, and RUN2 performed better on other ones. In the following, we will discuss why RUN1 and RUN2 did not perform equally on all topics. We will also discuss briefly the poor results of RUN3. We will not elaborate on RUN4, as it is composed of several models derived from runs 1 to 3.

A. Effect of the quality of Google Images results

We measured the precision of the results given by Google Images. The precision of the results given by Google Images is the number of relevant images divided by the total number of images. We found that there was no correlation between the precision of RUN1 and the precision of the images taken from Google Images, as one can see on the following figure.

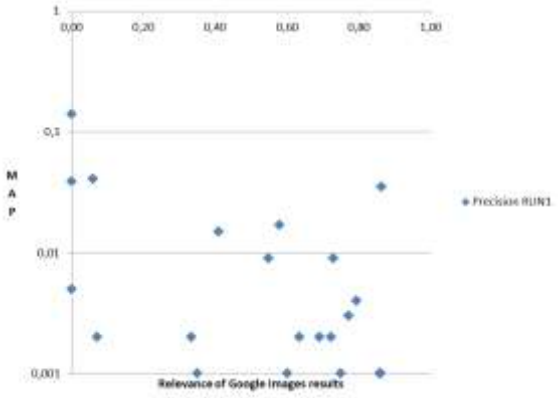

Figure 3 - MAP according to Relevance of Google Images results for RUN1

We also checked whether RUN1 performed better than other runs with better images. Again, we found no correlation.

B. Effect of the "viewable words rate"

We will say that a word in a topic is *viewable* if it gives a relevant visual information by itself within that specific topic. For instance in the topic "Find shots of a person playing drums indoors", viewable words are "person", "drums" and "indoors". In the topic "Find shots of sewing machines", there is no viewable word: neither "sewing" nor "machine" contain a relevant visual information by themselves. In the same way, in the topic "Find shots of a diver wearing diving suit and swimming under water", "diving" and "suit" are not viewable words. The concept of "viewable words" is not formal, but in most cases there is no ambiguity about what is viewable and what is not viewable. In the following array, the words that we manually identified as viewable are underlined for the first five topics. Note that some words such as "behind" and "under" provide some visual information, but only if they are put into context: that visual information is lost if words are considered separately.

Topic	Topic with viewable words underlined
1	Find shots of a person playing guitar outdoors
2	Find shots of a man indoors looking at camera where a bookcase is behind him
3	Find shots of a person playing drums indoors
4	Find shots of a diver wearing diving suit and swimming under water
5	Find shots of a person holding a poster on the street at daytime

Then for each topic, we computed a *viewable words rate* (VWR) by dividing the number of viewable words by the total number of words in the topic minus three (because we did not take into account "Find shots of" in our models). Eventually we plotted the curves of functions $f_1(t) = P(M_1 > M_2 \mid R > t)$, $f_2(t) = P(M_2 > M_1 \mid R > t)$, $f_3(t) = P(M_2 = M_1 \mid R > t)$, $g_1(t) = P(M_1 > M_2 \mid R < t)$, $g_2(t) = P(M_1 < M_2 \mid R < t)$ and $g_3(t) = P(M_1 = M_2 \mid R < t)$ with M_1 the MAP of RUN1, M_2 the MAP of RUN2 and R the viewable words rate. Here are the curves we obtained.

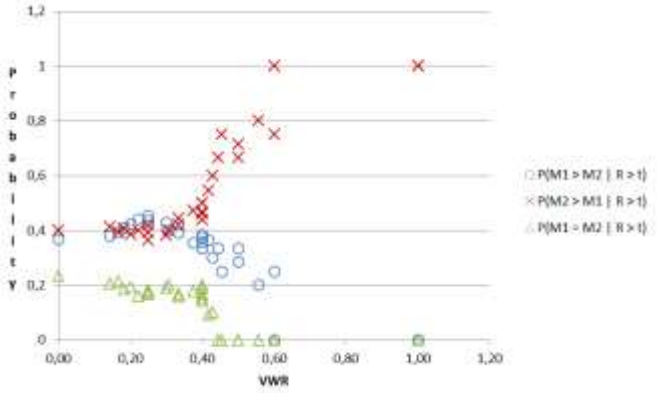

Figure 4 - Curves of f_1, f_2 and f_3

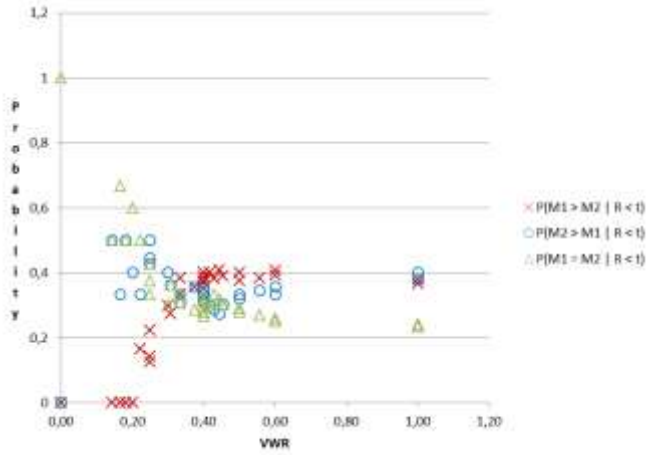

Figure 5 - Curves of g_1, g_2 and g_3

As one can see on the first plot, for VWR bigger than 0.40, the more the rate increases, the closer f_2 is to 1. In the second plot, neither g_1 nor g_2 become bigger than 0.5: it means that none of RUN1 and RUN2 perform better than the other on more than half of the topics for low VWR. The conclusion of these observations is that RUN2 performs much better than RUN1 for topics whose words are visually self-explanatory, whereas it performs similarly to RUN1 for other topics. As RUN2 relies on an average done word by word from semantic vectors generated by GloVe, it can be inferred that some visual information is lost by not putting words in context.

These observations can also explain the poor results of RUN3: as GloVe is run twice (on topics and on sentences generated by NeuralTalk), the phenomenon we described above is amplified.

V. Conclusion

In this paper we proposed two approaches for natural language access to video databases. In the first one we sent natural language queries to a web image search engine and compared results with keyframes. In the second one we generated text descriptions from keyframes and compared them with queries. We made four implementations using these approaches and compared them. We found that the best implementation among the four was using ImageShuffle to create feature vectors for keyframes and GloVe to compare images and sentences. We showed that this implementation of the second approach, had much better results than our implementation of the first approach for queries having what we called a high viewable words rate (VWR). We also found that the relevance of images taken from search engines had no impact on the performances of the first approach.

Thanks to these observations, we will now focus on queries with low VWR and explore methods to improve our implementation based on ImageShuffle and GloVe.

References

[1] George Awad, Jonathan Fiscus, Martial Michel, David Joy, Wessel Kraaij, Alan F. Smeaton, Georges Quénot, Maria Eskevich, Robin Aly, Roeland Ordelman, TRECVID 2016: Evaluating Video Search, Video Event Detection, Localization, and Hyperlinking, Proceedings of TRECVID 2016, 2016, NIST, USAJ. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] https://www.google.fr/imghp?

[3] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014.

[4] Pascal Mettes, Dennis C. Koelma, and Cees G.M. Snoek. 2016. The ImageNet Shuffle: Reorganized Pre-training for Video Event Detection. In Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval (ICMR '16). ACM, New York, NY, USA, 175-182.

[5] Andrej Karpathy, Li Fei-Fei, Deep Visual-Semantic Alignments for Generating Image Descriptions, CVPR 2015

[6] Jeffrey Pennington, Richard Socher, and Christopher D. Manning, GloVe: Global Vectors for Word Representation? Conference on Empirical Methods in Natural Language Processing, 2014

[7] Mikolov Tomáš, Karafiát Martin, Burget Lukáš, Černocký Jan, Khudanpur Sanjeev: Recurrent neural network based language model, In: Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010), Makuhari, Chiba, Japan

[8] S. Ayache and G. Quenot. Video Corpus Annotation using Active Learning. In European Conference on Information Retrieval (ECIR), pages 187–198, Glasgow, Scotland, mar 2008.

[9] Johnson, Justin and Karpathy, Andrej and Fei-Fei, Li , DenseCap: Fully Convolutional Localization Networks for Dense Captioning, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, Las Vegas, USA

[10] Amirhossein Habibian, Thomas Mensink, and Cees G. M. Snoek. 2014. Composite Concept Discovery for Zero-Shot Video Event Detection. In Proceedings of International Conference on Multimedia Retrieval (ICMR '14). ACM, New York, NY, USA.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc., 2012.

[12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Sys- tems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States., pages 3111–3119, 2013.

[13] U. Niaz, B. Merialdo, and C. Tanase. EURECOM at TRECVID 2014: The semantic indexing task. In TRECVID 2014, 18th International Workshop on Video Retrieval Evaluation, 10-12 November 2014, Orlando, USA, Orlando, UNITED STATES, 11 2014.

[14] Jeffrey Dalton, James Allan, and Pranav Mirajkar. 2013. Zero-shot video retrieval using content and concepts. In Proceedings of the 22nd ACM international conference on Conference on information & knowledge management (CIKM '13). ACM, New York, NY, USA

[15] B. Safadi, M. Sahuguet, and B. Huet. When Textual and Visual Information Join Forces for Multimedia Retrieval. In Proceedings of International Conference on Multimedia Retrieval (ICMR'14), pages 265:265–265:272, Glasgow, United Kingdom, 2014.