IRISA at TRECVid 2017: Beyond Crossmodal and Multimodal Models for Video Hyperlinking

Mikail Demirdelen, Mateusz Budnik, Gabriel Sargent,
Rémi Bois & Guillaume Gravier

IRISA, CNRS, Université de Rennes 1

firstname.lastname@irisa.fr

Abstract

This paper presents the runs that were submitted to the TRECVid Challenge 2017 for the Video Hyperlinking task. The goal of the task is to propose a list of video segments, called targets, to complement query video segments defined as anchors. The data provided with the task encourage participants to make use of multiple modalities such as the audio track and the keyframes. In this context, we submitted four runs: 1) *BiDNNFull* uses a BiDNN model to combine ResNet with Word2Vec; 2) *BiDNNFilter* makes use of the same model and also exploits the metadata to narrow down the list of possible candidates; 3) *BiDNNPinv* tries to improve on the anchor keyframe fusion by using the Moore-Penrose pseudo-inverse and finally 4) *noBiDNNPinv* tests on the relevance of not using a BiDNN to fuse the modalities. Our runs were built based on a pre-trained model of ResNet as well as the transcripts and the metadata provided by the organizers of the task. The results show a gain in performance over the baseline BiDNN model both when the metadata filter was used and when the keyframe fusion was done with a pseudo-inverse.

1 Introduction

The video hyperlinking task was once again revisited this year in TRECVid [ABF+17]. Its goal is to create links between different fragments of videos that share a similar topic. These links are established across a large and diverse collection of videos. Given a source video fragment, called an anchor, a list of video segments (or targets) is proposed. In the case of this evaluation, the anchors were manually created. And the systems proposed by participants have to automatically choose video segments relevant to a given anchor. This relevance or similarity should ideally be based on the semantic content of each video. In practice, relevance can often be expressed as a similarity criterion.

The challenge is to propose targets that can justifiably be linked to a given anchor and at the same time avoid redundancy. One of the ways to do it is to introduce a level of diversity that can offer unexpected yet relevant results.

The creation of hyperlinks consists of two main steps: a segmentation step, in which target candidate segments are extracted from the entire video dataset; and

a ranking step, in which the most relevant targets are selected for every anchor. The ranking is based on content analysis and different similarity measures. Both the segmentation and ranking are subject to many design decisions, each with its advantages and drawbacks.

For example, one could use a naive segmentation approach compensated by widely overlapping segments, or a more sophisticated segmentation allowing for the automatic removal of low-interest video fragments. In the first case, overlapping segments offer more opportunities to find a good matching video fragment for the anchor, while the second approach allows for less costly and faster comparisons at the ranking step due to fewer video pairs to compare. As for the ranking step, many aspects of the videos can be taken into account: from what is shown to what is said and from how it is said to how it is shown. Moreover, the Video Hyperlinking task provides many resources, including automatic transcripts, keyframes, visual concepts extracted from the keyframes and user-created metadata.

Our main goal in this year's submission is to compare multimodal and cross-modal approaches. We also want to improve the performance of these systems by exploring additional modalities (i.e. metadata) and experimenting with different fusion techniques for keyframes.

The rest of the paper is organized in the following way. Section 2 presents the data used in this evaluation and its representation. In Section 3 different runs are presented. Finally, Section 4 discusses some preliminary results.

2 Data and segmentation

The dataset used in this evaluation was the same as in 2016, namely the BlipTV dataset [AFM+16]. It contains 14 838 videos of a mean length of around 13 minutes. The videos present a variety of topics from computer science tutorials and sightseeing guides to homemade song covers. They are provided in many languages but a vast majority of them are in English. All the videos were used to train the models regardless of the language, while the anchor video fragments were exclusively in English.

Our submission to the task can be considered as an evolution of approaches that were proposed by our team last year [BVS+16] and, therefore, both have several key components in common. As in 2016, all of our four runs use the automatic transcripts that were provided by LIMSI [GLA02] to segment the input videos.

Last year's segmentation of the videos was expanded to improve coverage and provide more choices of potential targets. To do so, we chose to keep all the segments that last between 50 and 60 seconds (approximately half of the maximum authorized length for the segments) that did not cut the speech. This was implemented using a constraint programming framework. When there were no segments between 50 and 60 seconds in a video, the duration boundaries were extended to 10 and 120 seconds.

As a result, we obtained approximatively 1.1 million segments. As these segments were created naively and exhaustively, they often overlapped a lot. Nevertheless, they seem to provide better coverage and more choice than the segmentation used by our team last year (which was obtained following the procedure described in [BVS+16]). The latter segmentation was used as well as an

			Single		Average		Max	
Models	Layer	Dims	P@5	P@10	P@5	P@10	P@5	P@10
VGG19	FC8	1000	41.60	41.27	43.40	41.60	42.60	41.03
VGG19	FC7	4096	40.60	40.60	42.40	42.10	41.00	40.97
VGG19	FC6	4096	38.80	40.43	41.00	40.60	40.00	40.73
Inception	FC	1000	40.40	41.83	41.00	41.39	42.60	41.73
Inception	AP	1024	40.40	39.27	44.00	41.70	42.60	40.83
ResNext-101	AP	2048	41.00	39.37	41.40	40.10	41.80	39.90
ResNet-200	FC	1000	43.80	41.57	47.20	44.37	**47.60**	**44.87**
ResNet-200	AP	2048	42.00	41.30	44.80	43.20	43.80	43.10
ResNet-152	AP	2048	44.40	41.37	45.60	41.67	45.20	40.40

Table 1: Comparison of different visual descriptors using a cosine distance. Tests were made on the development set.

additional means to increase the number of target candidates. In total, around 1.4 million segments were used, which roughly is a 400% increase compared to last year.

Once the segments were ready, their transcripts and keyframes were extracted. Different experiments were made to choose the best vectorial representation for both the visual and the transcript modalities. These experiments as well as the preprocessing and the final embedding for each modality are described in detail in the 2 following subsections.

2.1 Visual representation

For the visual embeddings, several different deep convolutional neural network (CNN) architectures were tested as well as layers, from which the embeddings were obtained. The development anchor set was used in order to make the selection of a visual representation possible. This experiment was done using the annotations that were obtained last year. For each annotated target a single keyframe was extracted and subsequently embedded using different pre-trained CNN models. The same thing was done for the anchors on the development set. However in most cases, more than one image was used for each anchor. Each anchor had a list of annotated target candidates (both correct and not). A cosine distance measure was used to construct the ranking between the anchors in the development set and their corresponding annotated targets.

The results of these experiments can be seen in Table 1. Two evaluation measures were used: precision at 5 (P@5) and precision at 10 (P@10). Next to the name of the network, the name of the layer is shown from which the embedding was taken as well as the dimension of that embedding. There were either average pooling (AP) layers or fully connected layers (FC). A set of different state-of-the-art deep architectures were tested, including VGG19 [SZ14] (which was used in the previous evaluation in 2016), Inception [SLJ+15], two different versions of ResNet [HZRS16] and ResNext [XGD+16].

Because of the large number of segments, each target was represented by a single keyframe. Conversely, for the anchors on average 3 keyframes were used to describe each anchor segment. This gave the opportunity to test different fusion

techniques for the anchor representation. At this stage, 3 simple approaches were considered:

- Using a **single** keyframe and its vector representation while discarding the rest.
- Taking the **average** of all the vectors for a given anchor.
- Using the **maximum** value of each feature across the vectors for a given anchor.

Based on the results from Table 1, the fully connected layer from ResNet-200 was chosen along with the max representation for the anchor.

2.2 Transcript representation

The preprocessing of the transcripts consisted of a tokenization step. The model we used to create the textual embeddings was the same as last year, namely a `word2vec` skip-gram model with hierarchical sampling [MSC+13]. It uses a window size of 5 and produces a representation vector with a size of 100. Other continuous representations were tested such as `doc2vec` [LM14] and `skip-thought` [KZS+15]; however no improvements over `word2vec` were observed.

3 Our approach

Contrary to our choices in 2016, we decided to put more emphasis on the choice of visual descriptors, which seems to be a very important component in the overall performance. Moreover, the use of user-made metadata was explored and used in one of the runs. All of the above ideas are discussed in greater detail in the subsequent subsections.

3.1 Crossmodal Bidirectional Joint Learning - *BiDNN-Full*

In the first run, a bidirectional deep neural network (BiDNN) was trained with ResNet as a visual descriptor and a Word2Vec as a textual descriptor. This run is our baseline for testing other improvements to the model.

The BiDNN [VRG16] creates a crossmodal translation between two different modalities. This is done through the use of two separate neural networks for each translation while having the weights tied between the middle layer of each network. This should force the network to learn a common multimodal representation. Formally, the structure of the network can be defined as follows. Let $h_i^{(j)}$ denote the activation of a hidden layer at depth j in the network i (indicating one of the two modalities), x_i is the feature vector for a given modality i and y_i is the corresponding output of the network. The networks can be defined by their weight matrices $W_i^{(j)}$ and their bias vectors $b_i^{(j)}$ for each layer j. An activation function f is used to get the final output of each layer. The entire architecture (with 3 hidden layers where the middle one is the embedding) can be defined as follows:

$$h_i^{(1)} = f(W_i^{(1)} \times x_i + b_i^{(1)}) \qquad i = 1, 2 \tag{1}$$

$$h_1^{(2)} = f(W^{(2)} \times h_1^{(1)} + b_1^{(2)}) \tag{2}$$

$$h_1^{(3)} = f(W^{(3)} \times h_1^{(2)} + b_1^{(3)}) \tag{3}$$

$$h_2^{(2)} = f(W^{(3)T} \times h_2^{(1)} + b_2^{(2)}) \tag{4}$$

$$h_2^{(3)} = f(W^{(2)T} \times h_2^{(2)} + b_2^{(3)}) \tag{5}$$

$$o_i = f(W_i^{(4)} \times h_i^{(3)} + b_1^{(4)}) \tag{6}$$

The weight matrices $W^{(2)}$ and $W^{(3)}$ are used twice because of the weight tying. The input to the network is 1000 and 100 dimensions for the visual CNN feature vector and Word2Vec vector, respectively. The network is trained for 300 epochs using stochastic gradient descent with the learning rate of 0.1 and momentum 0.9. The tanh function was used as the activation function f as well as a dropout of 0.2. The output embedding is L2-normalized. After embedding both anchors and targets, the ranking is created based on a cosine distance.

3.2 BiDNN with metadata filter - *BiDNNFilter*

For the second run, in order to increase the precision of our system, we decided to use the metadata to filter out and narrow down the list of possible video candidates for each anchor. For each video, a file of metadata, provided by the organizers, contains various information like the title of the video, a short description, a list of tags, its license, informations about the uploader of the video, etc. This information was created by different uploaders and can, therefore, greatly vary in quality and quantity from one video to another.

The tags seemed to be a more relevant information source to use as it gives more precision about the topic of the video. Therefore, using a list of tags as a filter can allow our system to search for good targets only among the more relevant videos. To have an idea of the global quality of the tags, we computed some statistics on the dataset. We found that 77% of the videos had tags and that the average number of tags per video is 4.71. These numbers seemed quite low and could greatly reduce the number of target candidates to the point where there will be not enough diversity among them.

We chose to augment the size of the tag list by using the information in the descriptions as 86.6% of the videos had a description with a mean length of approximatively 40 words, stopwords excluded. To transform the written text of the descriptions to a list of keywords, we extracted verbs, nouns and adjectives and then went through a step of lemmatization followed by stopword and hapaxes removal. The augmented list of keywords – composed of the tags and the words from the description – brought more flexibility to our run.

The model of the second run is the same as in the previous one. The filter is added at the end of the pipeline where the system compares the vectorial representation of the anchor with the targets.

3.3 BiDNN with pseudo-inverse - *BiDNNPinv*

The third run tries to improve on the keyframe aggregation. For each anchor, several keyframes are extracted from a corresponding video segment. These images are aggregated to deal with different variations across the video segments. This was done using the Moore-Penrose pseudo-inverse, inspired by [SJ15] where it was shown that this approach can improve the search quality. This aggregation method called *memory vectors* can be defined in the following way. Given a set of anchor vectors represented as columns in a $d \times n$ matrix $X = [x_1, ..., x_n]$ where $x_i \in R^d$, the memory vector m can be described thanks to the Moore-Penrose pseudo-inverse X^+ as:

$$m(X) = (X^+)^T \mathbf{1}_n = X(X^T X)^{-1} \mathbf{1}_n \tag{7}$$

where $\mathbf{1}_n$ is a n dimensional vector with all values set to 1. This aggregation replaced the fusion techniques (average and max) presented in Section 2.1. It was later used alongside the transcript vector as an input to the BiDNN to create embedded crossmodal representations of the anchors. The ranking procedure follows the same steps as in the *BiDNNFull* run in Section 3.1.

3.4 Multimodal run with pseudo-inverse - *noBiDNNPinv*

Finally, the last run tests the usefulness of the BiDNN itself. That is, instead of using the BiDNN to create a crossmodal representation of two modalities, a concatenation of the vectors of each modality was used instead. This was done with the pseudo-inverse memory vectors (described in Section 3.3) and the transcript vectors from Word2Vec for the anchors. Considering the targets, the output of ResNet was used instead of the memory vector. The vectors were L2-normalized before concatenation. As before, the cosine distance was used to determine the ranking of the targets for each anchor.

4 Results

The evaluation of the task was done via Amazon's Mechanical Turker following the procedure described in [ELA+17]. The scores of our runs using different precision and MAP measures are shown in Table 2.

The run that uses the metadata—*BiDNNFilter*—obtained the best precision at 5 and 10, and the run that uses the pseudo-inverse keyframe fusion—*BiDNNPinv*—shows the best results for MAP and precision at 20. Moreover, we can notice that it also had the same precision at 10 as *BiDNNFilter* and a very close precision at 5.

Additionally, a significant difference between our runs *BiDNNPinv* and *noBiDNNPinv* can be observed showing the interest of using the BiDNN model for this task. However, considering the precision at 5 and 10, *noBiDNNPinv* performs better that our baseline *BiDNNFull*. We cannot conclude anything with absolute certainty – as both the vectorial representation and the keyframe fusion method change between the runs, but we can hypothesize that having a good keyframe fusion computation has more importance that just using a crossmodal neural network with a simple fusion method.

Runs	MAP	P@5	P@10	P@20
BiDNNFull	0.1334	0.6880	0.7120	0.4240
BiDNNFilter	0.1081	**0.7600**	**0.7440**	0.3800
BiDNNPinv	**0.1529**	0.7520	**0.7440**	**0.4340**
noBiDNNPinv	0.1246	0.7280	0.7320	0.3960

Table 2: The results for the 4 runs submitted using MAP, precision at 5, at 10 and 20.

5 Conclusion

This year's experiments validate our initial ideas about the importance of a keyframe fusion method and the relevance of a list of keyword filtering. It will be interesting to try and combine both of these points into a single model in order to see if it can outperform the other ones.

We also have shown that both the use of metadata and the use of BiDNN architectures are relevant and lead to better results. Therefore, trying to incorporate the metadata into the neural network architecture can be considered, using it as a potential third modality. It can however be risky as, contrary to the visual and textual representations, the metadata is available only on the video level. There could be a lot of redundancy if this data is used to train a model on the segment level.

References

[ABF+17] George Awad, Asad Butt, Jonathan Fiscus, David Joy, Andrew Delgado, Martial Michel, Alan F. Smeaton, Yvette Graham, Wessel Kraaij, Georges Quénot, Maria Eskevich, Roeland Ordelman, Gareth J. F. Jones, and Benoit Huet. Trecvid 2017: Evaluating ad-hoc and instance video search, events detection, video captioning and hyperlinking. In *Proceedings of TRECVID 2017*. NIST, USA, 2017.

[AFM+16] George Awad, Jonathan Fiscus, Martial Michel, David Joy, Wessel Kraaij, Alan F Smeaton, Georges Quénot, Maria Eskevich, Robin Aly, and Roeland Ordelman. Trecvid 2016: Evaluating video search, video event detection, localization, and hyperlinking. In *Proceedings of TRECVID*, 2016.

[BVS+16] Rémi Bois, Vedran Vukotić, Ronan Sicre, Christian Raymond, Guillaume Gravier, and Pascale Sébillot. Irisa at trecvid2016: Cross-modality, multimodality and monomodality for video hyperlinking. In *Proceedings of TRECVID*, 2016.

[ELA+17] Maria Eskevich, Martha Larson, Robin Aly, Serwah Sabetghadam, Gareth JF Jones, Roeland Ordelman, and Benoit Huet. Multimodal video-to-video linking: turning to the crowd for insight and evaluation. In *International Conference on Multimedia Modeling*, pages 280–292, 2017.

[GLA02] Jean-Luc Gauvain, Lori Lamel, and Gilles Adda. The limsi broadcast news transcription system. *Speech communication*, 37(1):89–108, 2002.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[KZS+15] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

[LM14] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1188–1196, 2014.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[SJ15] Ronan Sicre and Hervé Jégou. Memory vectors for particular object retrieval with multiple queries. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 479–482. ACM, 2015.

[SLJ+15] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[VRG16] Vedran Vukotić, Christian Raymond, and Guillaume Gravier. Bidirectional joint representation learning with symmetrical deep neural networks for multimodal and crossmodal applications. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 343–346. ACM, 2016.

[XGD+16] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016.