A study on multimodal video hyperlinking with visual aggregation

Mateusz Budnik, Mikail Demirdelen, Guillaume Gravier

A STUDY ON MULTIMODAL VIDEO HYPERLINKING WITH VISUAL AGGREGATION

Mateusz Budnik, Mikail Demirdelen and Guillaume Gravier

IRISA, CNRS, Université de Rennes 1, France
firstname.lastname@irisa.fr

ABSTRACT

Video hyperlinking offers a way to explore a video collection, making use of links that connect segments having related content. Hyperlinking systems thus seek to automatically create links by connecting given anchor segments to relevant targets within the collection. In this paper, we further investigate multimodal representations of video segments in a hyperlinking system based on bidirectional deep neural networks, which achieved state-of-the-art results in the TRECVid 2016 evaluation. A systematic study of different input representations is done with a focus on the aggregation of the representation of multiple keyframes. This includes, in particular, the use of memory vectors as a novel aggregation technique, which provides a significant improvement over other aggregation methods on the final hyperlinking task. Additionally, the use of metadata is investigated leading to increased performance and lower computational requirements for the system.

Index Terms— Video hyperlinking, multimodal embedding, similarity search, memory vectors

1. INTRODUCTION

Video hyperlinking appeared recently as the task of organizing large video collections with explicit links between related video fragments. Hyperlinking can be seen as a complement to video retrieval, focusing on the browsing and exploration stages that follow the search: after finding entry points of interest within the collection, users can browse and explore following the links created by the video hyperlinking process [1]. Over the past years, video hyperlinking has been implemented and evaluated in yearly international benchmarks, first within the MediaEval initiative [2], then as part of the TRECVid series of evaluations [3].

Video hyperlinking systems usually start from a set of *anchors* that define entry points of interest in collections of long videos and are required to provide, for each anchor, relevant targets within the collection. This task is usually implemented as a two-step process, first starting from a segmentation of the long videos into small segments, then selecting relevant segments for a given anchor [4, 5]. This last step is cast as a video retrieval task relying on video segment comparison, where various multimodal solutions have been proposed. For instance, in the 2016 TRECVid evaluation, many teams investigated late fusion approaches combining transcript-based, visual-based and metadata-based target retrieval systems. The state of the art was, however, achieved with multimodal representation based on bidirectional neural networks (BiDNNs) embedding transcripts and keyframes in a common representation space attainable from text embeddings with average word2vec and/or from image embeddings with the VGG-19 convolutional neural network (CNN) [6].

While the video hyperlinking task can be cast as a video retrieval task after video segmentation, a major difference is the use of video fragments as queries in hyperlinking. Most of the work in video retrieval focuses on textual or image queries (cross-modal approaches) [7, 8], or on near duplicate video retrieval [9]. In hyperlinking, we rather seek to create multimodal embeddings of video fragments to group and link relevant segments together. Most methods used in the video hyperlinking task over the past few years indeed rely on multimodal representations to find relevant video fragments, with joint embeddings performing best [3].

In this paper, we further improve on the BiDNN approach to target selection in video hyperlinking with a systematic study of various input representations. On the one hand, BiDNNs as implemented in the TRECVid 2016 evaluation does not make use of the latest advances in CNN-based features. In [10], authors observe a significant gain after switching from AlexNet to VGG architectures. We thus investigate the impact of changing the representation of the keyframes of each segment to recent very deep CNN architectures such as ResNet [11], Inception [12] and ResNext [13]. At the transcript level, document and LSTM-based embeddings of texts are considered as a replacement to the average word2vec representation. At the visual modality level, we investigate aggregation of the representation of the multiple keyframes that depict a video segment to account for the temporal structure of videos. In addition to pooling strategies such as average or max pooling, we consider a recently introduced technique, namely memory vectors [14], which enables the aggregation of multiple descriptors into a single one, yet maintaining good properties for information retrieval. Memory vectors were successfully used as building blocks for efficient indexing structures in image retrieval. We demonstrate in this paper their ability to act as feature aggregators and show their ben-

efit in a video hyperlinking task.

The paper is organized as follows. We first describe the global architecture of the full video hyperlinking system and detail the aggregation components that we propose and compare. We then describe the data and the experimental setup, before reporting and commenting on the results in a target reranking setting to assess the benefit of the various representation and aggregation techniques compared. Based on these results, we describe a full video hyperlinking system and assess it in the TRECVid 2017 video hyperlinking task.

2. AGGREGATION METHODS IN VIDEO HYPERLINKING

Video hyperlinking systems implement a complete pipeline to turn a collection of raw videos with a given set of anchors, i.e., pre-defined entry points in the collection, into a structured collection with a number of hyperlinks departing from the pre-defined set of anchors. At the core of the system is a content comparison algorithm that builds on a multimodal embedding of video fragments that are to be compared, where multimodal embedding is obtained by means of a BiDNN. As multiple keyframes are present in the video fragments to be compared, one needs to aggregate the visual representations over different keyframes in order to use BiDNNs. After a brief overview of the general content matching mechanism that we use, we present the various aggregation strategies considered in this paper.

2.1. System overview

The video hyperlinking system starts by segmenting videos into small segments which are seen as potential targets to select from for a given anchor. For retrieval purposes, each segment is represented in a multimodal representation space in which segments will be compared to anchors using standard retrieval methods.

Different steps towards the multimodal embedded representation of a segment are illustrated in Figure 1. The video frames and the speech transcript are extracted and, for each modality, a unique embedding is built for further projection in a multimodal representation space using BiDNN. For the visual modality, a descriptor of each frame is obtained by passing the image through a pretrained convolutional neural network. When several frames are present in a video segment, an aggregation step is performed to produce a single vector representation that will serve as input to the BiDNN fusion. Regarding the audio modality, the transcript of the video segment is embedded to produce the second input to the BiDNN fusion. Different embeddings of transcripts and images are compared in the experiments.

Fusion of the two modalities is performed with a BiDNN [6], a model which creates a crossmodal translation between two modalities. This is done through the use of two

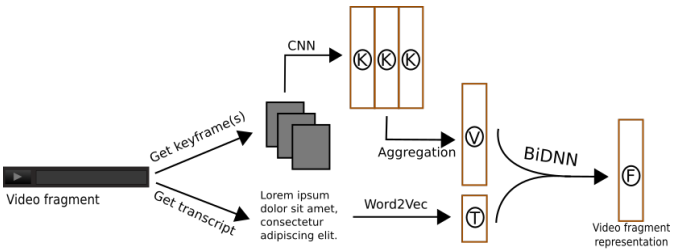

Fig. 1. The overview of the system including the extraction of different modalities.

separate neural networks for each translation while having the weights tied between the middle layer of each network. This forces the network to learn a common multimodal representation. Formally, the structure of the network is given by

$$\mathbf{h}_i^{(1)} = f(\mathbf{W}_i^{(1)} \times \mathbf{x}_i + \mathbf{b}_i^{(1)}) \quad i = 1, 2 \tag{1}$$

$$\mathbf{h}_1^{(2)} = f(\mathbf{W}^{(2)} \times \mathbf{h}_1^{(1)} + \mathbf{b}_1^{(2)}) \tag{2}$$

$$\mathbf{h}_1^{(3)} = f(\mathbf{W}^{(3)} \times \mathbf{h}_1^{(2)} + \mathbf{b}_1^{(3)}) \tag{3}$$

$$\mathbf{h}_2^{(2)} = f(\mathbf{W}^{(3)T} \times \mathbf{h}_2^{(1)} + \mathbf{b}_2^{(2)}) \tag{4}$$

$$\mathbf{h}_2^{(3)} = f(\mathbf{W}^{(2)T} \times \mathbf{h}_2^{(2)} + \mathbf{b}_2^{(3)}) \tag{5}$$

$$\mathbf{o}_i = f(\mathbf{W}_i^{(4)} \times \mathbf{h}_i^{(3)} + \mathbf{b}_1^{(4)}) \tag{6}$$

where $\mathbf{h}_i^{(j)}$ denotes the activation of a hidden layer at depth j in the network i (indicating one of the two modalities), $\mathbf{x}_i$ is the feature vector for a given modality i and $\mathbf{y}_i$ is the corresponding output of the network. The parameters are the weight matrices $\mathbf{W}_i^{(j)}$ and the bias vectors $\mathbf{b}_i^{(j)}$ and function f is a ReLU function. Training seeks to minimize the mean square error of $(\mathbf{x}_1, \mathbf{o}_2)$ and $(\mathbf{x}_2, \mathbf{o}_1)$.

2.2. Aggregation

In the case of anchor video segments, usually several keyframes are extracted. In order to have a single vector representation, an aggregation step is performed. These image representations are fused to deal with different variations across the video segment. Four aggregation approaches were considered and they can be defined as follows. Let $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n]$, where $\mathbf{x}_i \in R^d$, be a $d \times n$ matrix representing anchor vectors. A simple average can be used to create an aggregate vector $\mathbf{a}$:

$$\mathbf{a}(\mathbf{X}) = \frac{1}{n}\mathbf{X}\mathbf{1}_n \tag{7}$$

where $\mathbf{1}_n$ is a n dimensional vector with all values set to 1. The second approach uses a maximum response for each corresponding value in the vectors:

$$\mathbf{a}(\mathbf{X}) = \max_{j \in D} x_{ji}, \quad i = 1, \ldots, n \tag{8}$$

where $D = \{1, \ldots, d\}$ and x is a corresponding value in matrix $\mathbf{X}$. The third approach selects a just single vector and discards the rest:

$$\mathbf{a}(\mathbf{X}) = \mathbf{x}_i \tag{9}$$

In the case of our experiments, the vector of the first keyframe in the video fragment was selected as its representative.

Memory vectors

The last aggregation approach was done using the Moore-Penrose pseudo-inverse. This aggregation method called *memory vectors* can be defined in the following way thanks to the Moore-Penrose pseudo-inverse $\mathbf{X}^+$ as:

$$\mathbf{a}(\mathbf{X}) = (\mathbf{X}^+)^T \mathbf{1}_n = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{1}_n \tag{10}$$

The name 'memory vector' refers to the ability to summarize a set of vectors into a single vector, while maintaining the distinctiveness of each member vector. This method was initially used to aggregate multiple image descriptors into a single vector representation to speed up image retrieval [14]. In this case, the optimal vector representation with respect to the false positive rate is obtained by maximizing the membership score over all member vectors while minimizing the memory vector norm. The solution to this optimization problem can be given by the pseudo-inverse, considering stricter constraints to eliminate interferences between vectors gathered in the same 'memory' and yielding a better balance between rare and frequent features. In this work, we propose to use memory vectors at the video level.

These aggregation techniques are evaluated and the results presented in Section 4.2. The $\mathbf{a}$ vector was later used alongside the transcript vector as an input to the BiDNN to create embedded crossmodal representations of the anchors.

3. DATA AND EXPERIMENTAL SETUP

The experiments were carried out on the BlipTV dataset [3]. 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.

Additionally, many other sources of information are available for each video: multiple transcripts generated by the LIMSI2016 [15] speech-to-text system, the timecodes of the shots, keyframes that correspond to the middle frame of each shot and the metadata. The latter contains information on the video provided by the author such as its title, a description of the video, a list of tags that mainly describe its semantic content, its license, information about the author, etc.

Models	P@5	MAP
Average Word2Vec	**44.2**	**45.3**
Doc2Vec	38.4	39.4
Skip-Thought	40.2	41.6

Table 1. Results for the textual descriptor evaluation.

4. A STUDY ON VISUAL AGGREGATION

In this section, the experiments for the modality representation are introduced with emphasis on the visual modality and its aggregation. The evaluation was carried out on a reranking task composed of the development anchors of the BlipTV dataset. Each anchor had a list of annotated target candidates (both correct and incorrect). The annotations were obtained during the 2016 TRECVid Hyperlinking evaluation.

The development set is composed of 89 anchors. The total number of annotated video segments is 7216, which gives on average around 81 segments per anchor. The annotation is binary (relevant or not relevant for a given anchor). All the tests in this section were made using this dataset.

4.1. Textual representation

To represent the speech of the videos, the transcripts produced by LIMSI2016 were incorporated. Three different textual neural networks were tested: an averaged word2vec[16] on each words of the segment, doc2vec[17] and skip-thought[18]. The procedure was the same as for the visual features, except the measures used were precision at 5 (P@5) and mean averaged precision (MAP). The results are shown in Table 1. Despite carefully tuning the parameters, the standard word2vec outperformed its the newer alternatives. Therefore, an averaged word2vec was chosen for the text representation.

4.2. 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. 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 in the development set. However, more than one image was used for each anchor. The fusion of the anchor vectors was done through average aggregation, which is a well established approach that can provide stable results [10]. 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 2. 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 [19], Inception [12], two different versions of ResNet [11] and ResNext [13].

			Average	
Models	Layers	Dims	P@5	P@10
VGG19	FC8	1000	43.40	41.60
VGG19	FC7	4096	42.40	42.10
VGG19	FC6	4096	41.00	40.60
Inception	FC	1000	41.00	41.39
ResNext-101	AP	2048	41.40	40.10
ResNet-200	FC	1000	**47.20**	**44.37**
ResNet-200	AP	2048	44.80	43.20
ResNet-152	AP	2048	45.60	41.67

Table 2. Results for the visual descriptor evaluation. The average aggregation was used.

Based on the results in Table 2, the ResNet-200 descriptors give the highest precision. The last fully connected layer was chosen as the base visual descriptor for the hyperlinking evaluation. It seems that the overall performance of a given descriptor in this case is linked to its semantic meaning: in both the VGG19 and ResNet-200 the top layers (which indicate presence of a given concept in an image) are outperforming other layers further down. Even though the latter are considered more suitable for more general representation [20].

4.3. Aggregation over keyframes

In this section, the performance of different aggregation techniques introduced in Section 2.2 is presented. Table 3 provides the results based on the ResNet-200 descriptors, which displayed the highest precision in the evalution as shown in the previous section. Based on the results, it seems that max aggregation gives the best overall results for P@5 and P@10. Because of that it was chosen as a baseline in the other complete system evaluation. Despite its lower scores for the ResNet-200 fully connected layer, the memory vectors give the highest performance for P@1 and outperform the max aggregation at P@5 for the other descriptor. Because of the above and keeping in mind the limitations of this evaluation, the memory vectors were chosen to be also used in the hyperlinking evaluation.

5. COMPLETE SYSTEM EVALUATION

Despite seemingly interesting results on the reranking evaluation, this set is far from a complete system and the results must be taken with hindsight. In order to have more accurate and representative results, we used the test set given in the 2017 edition of the TRECVid video hyperlinking task. In this set, the evaluation is done on 25 anchors for which we can make segment suggestion from the whole database and whose relevance was evaluated by human assessors.

	ResNet-200 FC			ResNet-200 AP		
Agg.	P@1	P@5	P@10	P@1	P@5	P@10
Single	44.00	43.80	41.57	43.00	42.00	41.30
Avg	45.00	47.20	44.37	47.00	44.80	43.20
Max	48.00	**47.60**	**44.87**	44.00	43.80	43.10
Mem_v	45.00	44.20	43.30	**52.00**	44.40	41.50

Table 3. Results for the visual aggregation evaluation based on the ResNet-200 descriptors.

5.1. Segmentation

As stated before, a first step of the segmentation was done to fragment each video into segments that can better represent a particular topic or context at a given moment of the video. The difficulty of this step is to have a good coverage of each video without being too redundant. Moreover, no segment should cut the speech. Two methods of segmentation were used in order to cover both the number and the quality of the segments. The first method is a sliding window applied to each video that creates segments that have 30 seconds of continuous speech. This process is applied a second time with an offset with already found segments as described in [21].

While the above method is good to have control over the segment we produce, it scales badly. In order to produce a large number of segments, a constraint programming framework was used. The length of segments was fixed to be between 50 and 60 seconds. When this was too restrictive, it was expanded to between 10 and 120 seconds. The first method that was used gave around 300.000 segments and the second produced an additional 1.1 million segments, to a total of 1.4 million segments.

5.2. Metadata

To augment the precision of our system, the metadata associated with the videos was tested as a way to filter out and narrow down the list of possible target candidates. The list of tags seemed particularly relevant to serve as a filter. However, only 77% of the videos have tags and with a mean number of 4.71 tags per videos. These numbers seemed too restrictive and might result in an overfiltering of the results.

The list of tags was expanded using the text of the video descriptions that are present in 86.6% of the videos with a mean number of 39.8 words excluding stopwords. From these descriptions, the nouns, verbs and adjectives were extracted, lemmatized, afterwards the stopwords and hapaxes were removed. This list of keywords—tags and descriptions—was

kept and used to select only videos that share at least one word with the anchor. It is worth noting that the filtering was done on the complete video level. The particular segment within a video still needs to be selected.

5.3. Experimental setup

Four different approaches and variations are compared and evaluated:

- *BiDNNFull* represents the BiDNN algorithm with ResNet-200 visual features and max aggregation (for anchors only). This approach serves as the baseline.
- *BiDNNFilter* the exact same setup as above. However, the metadata is used as a filter as described in Section 5.2.
- *BiDNNMemVec* replaces the max aggregation with a memory vector for the visual representation for anchors. All the other parameters are analogues to *BiDNNFull*. The memory vector representation is embedded using the BiDNN algorithm.
- *noBiDNNMemVec* does not use the BiDNN model, instead, a simple concatenation of the two modalities is used. For the visual representation, the memory vector is used for aggregation of the anchor segments.

The BiDNN-based system was trained with stochastic gradient descent, 0.9 momentum and 20% dropout for 300 epochs (even though it seemed to converge earlier than that). The input video and audio representations had of the size 1000 and 100, respectively. The resulting embedding was an L2 normalized 1024-dimensional vector. Cosine distance was used for matching the embedded vectors.

5.4. Results and discussion

The scores of the hyperlinking evaluation are given in Table 4 using the precision at 5, 10 and 20 (denoted as P@5, P@10 and P@20, respectively) as well as the mean average precision (MAP).

The approach using the memory vectors (*BiDNNMemVec*) seems to outperform the baseline at every measure, while having the biggest difference at P@5. It is important to note that both systems use the same trained BiDNN model, so the improvement is due to the use of memory vectors over max aggregation.

The use of the metadata (*BiDNNFilter*) led to surprisingly good results giving the best precision at 5 out of any approach and comparable result for P@10. The drawback seems to be that too many correct potential targets are filtered out. This can be seen when considering P@20 and MAP scores, which take more targets into account. The *BiDNNMemVec* seems to handle this problem better, showing more stability of the quality of the results across different metrics.

Approach	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
BiDNNMemVec	**0.1529**	0.7520	**0.7440**	**0.4340**
noBiDNNMemVec	0.1246	0.7280	0.7320	0.3960

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

Additionally, a significant difference between the methods: *BiDNNMemVec* and *noBiDNNMemVec* can be observed showing the interest of using the BiDNN model for this task. However, considering the precision at 5 and 10, *noBiDNNMemVec* performs better than the *BiDNNFull* baseline. This seems to indicate the importance of the choice of the input representation on the overall performance of the system.

6. CONCLUSION

In this paper, a study on the performance of different visual representations was presented as well as several aggregation methods. The results show that the initial choice of the representation has a significant effect on the overall performance of the system. The same is true for the choice of the aggregation approach, especially in the case of the memory vectors, which seem to provide significant gains on the hyperlinking task. In this study the aggregation was performed only on the anchor side. Therefore, one way of potential improvement could be to extract multiple keyframes for each segments, to have a better representativity of the whole segments. However, doing so could be computationally expensive.

It was also 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.

7. REFERENCES

[1] Maria Eskevich, Gareth JF Jones, Robin Aly, Roeland JF Ordelman, Shu Chen, Danish Nadeem, Camille Guinaudeau, Guillaume Gravier, Pascale Sébillot, Tom De Nies, et al., "Multimedia information seeking through search and hyperlinking," in *Proceedings of the 3rd ACM conference on International conference on multimedia retrieval*. ACM, 2013, pp. 287–294.

[2] Maria Eskevich, Robin Aly, David Racca, Roeland Ordelman, Shu Chen, and Gareth JF Jones, "The search and hyperlinking task at mediaeval 2014," 2014.

[3] 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.

[4] Evlampios Apostolidis, Vasileios Mezaris, Mathilde Sahuguet, Benoit Huet, Barbora Červenková, Daniel Stein, Stefan Eickeler, José Luis Redondo Garcia, Raphaël Troncy, and Lukás Pikora, "Automatic fine-grained hyperlinking of videos within a closed collection using scene segmentation," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 1033–1036.

[5] Petra Galuščáková, Michal Batko, Jan Čech, Jiří Matas, David Novák, and Pavel Pecina, "Visual descriptors in methods for video hyperlinking," in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*. ACM, 2017, pp. 294–300.

[6] 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*. ACM, 2016, pp. 343–346.

[7] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim, "End-to-end concept word detection for video captioning, retrieval, and question answering," in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, 2017, pp. 3261–3269.

[8] André Araujo, Jason Chaves, Roland Angst, and Bernd Girod, "Temporal aggregation for large-scale query-by-image video retrieval," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1519–1522.

[9] Xiao Wu, Alexander G Hauptmann, and Chong-Wah Ngo, "Practical elimination of near-duplicates from web video search," in *Proceedings of the 15th ACM international conference on Multimedia*. ACM, 2007, pp. 218–227.

[10] Vedran Vukotić, Christian Raymond, and Guillaume Gravier, "Multimodal and crossmodal representation learning from textual and visual features with bidirectional deep neural networks for video hyperlinking," in *Proceedings of the 2016 ACM workshop on Vision and Language Integration Meets Multimedia Fusion*. ACM, 2016, pp. 37–44.

[11] 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*, 2016, pp. 770–778.

[12] 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*, 2015, pp. 1–9.

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

[14] Ahmet Iscen, Teddy Furon, Vincent Gripon, Michael Rabbat, and Hervé Jégou, "Memory vectors for similarity search in high-dimensional spaces," *IEEE Transactions on Big Data*, 2017.

[15] Jean-Luc Gauvain, Lori Lamel, and Gilles Adda, "The limsi broadcast news transcription system," *Speech communication*, vol. 37, no. 1, pp. 89–108, 2002.

[16] 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*, 2013, pp. 3111–3119.

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

[18] 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*, 2015, pp. 3294–3302.

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

[20] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[21] Rémi Bois, Vedran Vukotić, Ronan Sicre, Christian Raymond, Guillaume Gravier, and Pascale Sébillot, "Irisa at trecvid2016: Crossmodality, multimodality and monomodality for video hyperlinking," in *Proceedings of TRECVID*, 2016.