Learning a Text-Video Embedding from Incomplete and Heterogeneous Data

Antoine Miech[1,2] Ivan Laptev[1,2] Josef Sivic[1,2,3]

[1]École Normale Supérieure[*] [2]Inria [3]CIIRC[**]
http://www.di.ens.fr/willow/research/mee/

Abstract. Joint understanding of video and language is an active research area with many applications. Prior work in this domain typically relies on learning text-video embeddings. One difficulty with this approach, however, is the lack of large-scale annotated video-caption datasets for training. To address this issue, we aim at learning text-video embeddings from heterogeneous data sources. To this end, we propose a Mixture-of-Embedding-Experts (MEE) model with ability to handle missing input modalities during training. As a result, our framework can learn improved text-video embeddings simultaneously from image and video datasets. We also show the generalization of MEE to other input modalities such as face descriptors. We evaluate our method on the task of video retrieval and report results for the MPII Movie Description and MSR-VTT datasets. The proposed MEE model demonstrates significant improvements and outperforms previously reported methods on both text-to-video and video-to-text retrieval tasks.
Code: https://github.com/antoine77340/Mixture-of-Embedding-Experts

1 Introduction

Automatic video understanding is an active research topic with a wide range of applications including activity capture and recognition, video search, editing and description, video summarization and surveillance. In particular, the joint understanding of video and natural language holds a promise to provide a convenient interface and to facilitate access to large amounts of video data. Towards this goal recent works study representations of vision and language addressing tasks such as visual question answering [1, 2], action learning and discovery [3–5], text-based event localization [6] as well as video captioning, retrieval and summarization [7–10]. Notably, many of these works adopt and learn joint text-video representations where semantically similar video and text samples are mapped to close points in the *joint embedding space*. Such representations have been proven efficient for joint text-video modeling e.g., in [4, 6–9].

Learning video representations is known to require large amounts of training data [11, 12]. While video data with label annotations is already scarce, obtaining a large number of videos with text descriptions is even more difficult. Currently available video

[*] [1]WILLOW project, Département d'informatique, École Normale Supérieure, CNRS, PSL Research University, 75005 Paris, France.

[**] [3]Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague.

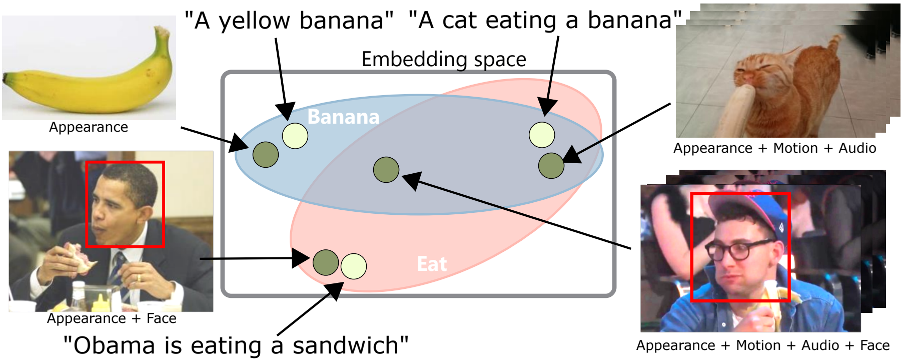

Fig. 1: We learn a text-video embedding from diverse and partially available data sources. This example illustrates a joint text-video embedding trained from videos and images while combining descriptors for global appearance, motion, audio and faces. The key advantage of our method is the ability to combine samples with different subsets of modalities, e.g., images with no motion and videos with no faces or sound.

datasets with ground truth captions include DiDeMo [6] (27K unique videos), MSR-VTT [13] (10K unique videos) and the MPII Movie Description dataset [14] (120K unique videos). To compensate for the lack of video data, one possibility would be to pre-train visual representations on still image datasets [12] with object labels or image captions such as ImageNet [15], COCO [16], Visual Genome [17] and Flickr30k [18]. Pre-training, however, does not provide a principled way of learning from different data sources and suffers from the "forgetting effect" where the knowledge acquired from still images is removed during fine-tuning on video tasks. More generally, it would be beneficial to have methods that can learn embeddings simultaneously from heterogeneous and partially-available data sources such as appearance, motion and sound but also from other modalities such as facial expressions or human poses.

In this work we address the challenge of learning from heterogeneous data sources. Our method is designed to learn a joint text-video embedding and is able to handle missing video modalities during training. To enable this property, we propose a Mixture-of-Embedding-Experts (MEE) model that computes similarities between text and a varying number of video modalities. The model is learned end-to-end and generates expert weights determining individual contributions of each modality. During training we combine image-caption and video-caption datasets and treat images as a special case of videos without motion and sound. For example, our method can learn an embedding for "Eating banana" even if "banana" only appears in training images but never in training videos (see Fig. 1). We evaluate our method on the task of video retrieval and report results for the MPII Movie Description and MSR-VTT datasets. The proposed MEE model demonstrates significant improvements and outperforms all previously reported methods on both text-to-video and video-to-text retrieval tasks.

Our MEE model can be easily extended to other data sources beyond global appearance, motion and sound. In particular, faces in video contain valuable information including emotions, gender, age and identities of people. As not all videos contain people, faces constitute a typical case of a potentially missing data source for our model. To demonstrate the generalization of our model and to show the importance of faces for video retrieval, we compute facial descriptors for images and videos with faces. We then treat faces as an additional data source in the MEE model and aggregate facial descriptors within a video (see Fig. 2). The resulting MEE combining faces with appearance, motion and sound produces consistent improvements in our experiments.

1.1 Contributions

This paper provides the following contributions: *(i)* First, we propose a new model for learning a joint text-video embedding called Mixture-of-Embedding-Experts (MEE). The model is designed to handle missing video modalities during training and enables simultaneous learning from heterogeneous data sources. *(ii)* We showcase two applications of our framework. First, we can data augment video-caption datasets with image-caption datasets during training. We can also leverage face descriptors in videos to improve the joint text-video embedding. In both cases, we show improvements in several video retrieval benchmarks. *(iii)* By using MEE and leveraging multiple sources of training data we outperform state-of-the-art on the standard text-to-video and video-to-text retrieval benchmarks defined by the LSMDC [14] challenge.

2 Related work

In this section we review prior work related to vision and language, video representations and learning from sources with missing data.

2.1 Vision and Language

There is a large amount of work leveraging language in computer vision. Language is often used as a more powerful and subtle source of supervision than predefined classes. One way to leverage language in vision is to find a joint embedding space for both visual and textual modalities [7–9, 19–23]. In this common embedding space, visual and textual samples are close if and only if they are semantically similar. This common embedding space enables multiple applications such as text-to-image/video retrieval and image/video-to-text retrieval. The work of Aytar *et al.* [24] is going further by learning a cross-modal embedding space for visual, textual and aural samples. In vision, language is also used in captioning where the task is to generate a descriptive caption of an image or a video [10, 25–27]. Another related application is visual question answering [1, 2, 28, 29]. A useful application of learning jointly from video and text is the possibility of performing video summarization with natural language [8]. Other works also tackle the problem of visual grounding of sentences: it can be applied to spatial grounding in images [18, 25, 30] or temporal grounding (*i.e* temporal localization) in videos [4, 6]. Our method improves text-video embeddings and has potential to improve any method relying on such representations.

2.2 Multi-stream video representation

Combining different modalities is a straightforward way to improve video representations for many tasks. Most state-of-the-art video representations [31–35] separate videos into multiple stream of modalities. The appearance, which are features capturing visual cues, the motion, computed from optical flow estimation or dense trajectories [36], and the audio signal are the commonly used video modalities. Investigating on which video descriptors to combine and how to efficiently fuse them has been extensively studied. Most prior works [12, 31, 33, 37, 38] address the problem of appearance and motion fusion for video representation. Other more recent works [34, 39] explore appearance-audio two-stream architectures for video representation. This and other work has consistently demonstrated the benefits of combining different video modalities for tasks such as video classification and action recognition. Similar to previous work in video understanding, our model combines multiple modalities but can also handle missing modalities during training and testing.

2.3 Learning with missing data

Our work is also closely related to learning methods designed to handle missing data. Handling missing data in machine learning is far from being a solved problem, yet it is widespread in various fields. Data can be missing due to several reasons: it can be corrupted, it may have not been possible to record the data or, in some cases, the data may be intentionally missing (take an example of forms with answers to some fields being optional). Common practices in machine learning aim at imputing the missing values with a default value such as zero, the mean, the median or the most frequent value in the discrete case[1]. In the matrix completion theory, a low rank approximation of the matrix [40] can be performed to fill the missing values. In computer vision, one main application of learning with missing data is the inpainting task. Several approaches such as: Low rank matrix factorization [41], Generative Adversarial Network [42] or more recently [43] have successfully addressed the problem. The UberNet network [44] is a universal multi-task model aiming at solving multiple problems such as: object detection, object segmentation or surface normal estimation. To do so, the model is trained on a mix of different annotated datasets, each one having its own task-oriented set of annotation. Their work is also related to ours as we also combine diverse types of datasets. However in our case, we have to address the problem of missing video modalities instead of missing task annotation.

Handling missing modalities can be seen as a specific case of learning from missing data. In image recognition the recent work [45] has tackled the task of learning with missing modalities to treat the problem of missing sensor information. In this work, we address the problem of missing video modalities. As explained above, videos can be divided into multiple relevant modalities such as appearance, audio and motion. Being able to train and infer models without all modalities makes it possible to mix different type of data such as illustrated in Figure 1.

[1] `http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.Imputer.html`

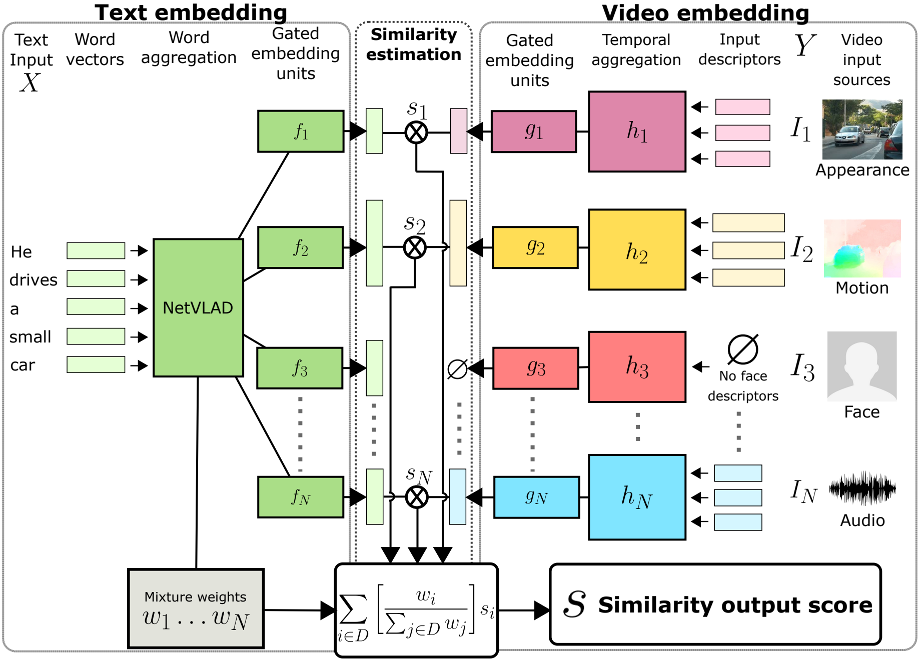

Fig. 2: **Mixture of embedding experts (MEE) model** that computes similarity score s between input sentence X and video Y as a weighted combination of expert embeddings, one for each input descriptor type including appearance, motion, facial descriptors or audio. The appropriate weight of each expert is estimated from the input text. Our model can deal with missing video input such as face descriptors missing for videos without people depicted above.

3 Mixture of embedding experts for video and text

In this section we introduce the proposed mixture of embedding experts (MEE) model and explain how this model handles heterogeneous input sources with incomplete sets of data streams during both training and inference.

3.1 Model overview and notation

Our goal is to learn a common embedding space for video and text. More formally, if X is a sentence and Y a video, we would like to learn embedding functions f and g such that similarity $s = \langle f(X), g(Y) \rangle$ is high if and only if X and Y are semantically similar. We assume that each input video is composed of N different streams of descriptors, $\{I_i\}_{i \in 1 \dots N}$ that represent, for example, motion, appearance, audio, or facial appearance of people. Note that as we assume the videos come from diverse data sources a particular video may contain only a subset of these descriptor types. For example, some videos may not have audio, or will not have face descriptors when they don't depict people. As we will show later, the same model will be able to represent still images as (very) simple videos composed of a single frame without motion. To address the issue that not all videos will have all descriptors, we design a model inspired by the mixture of experts [46], where we learn a *separate* "expert" embedding model for each descriptor type. The expert embeddings are combined in an end-to-end trainable fashion using weights that depend on the input caption. As a result, the model can learn to increase the relative weight of motion descriptors for input captions concerning human actions, or increase the relative weight of face descriptors for input captions that require detailed face understanding.

The overview of the model is shown in Figure 2. Descriptors of each input stream I_i are first aggregated over time using the temporal aggregation module h_i and the resulting aggregated descriptor is embedded using a gated embedding module g_i (see 3.4). Similarly, the individual word embeddings from the input caption are first aggregated using a text aggregation module into a single descriptor, which is then embedded using gated embedding modules f_i, one for each input source i. The resulting expert embeddings for each input source are then weighted using normalized weights $w_i(X)$ estimated by the weight estimation module from caption X to obtain the final similarity score s. Details of the individual components are given next.

3.2 Text representation

The textual input is a sequence of word embeddings for each input sentence. These individual word embedding vectors are then aggregated into a single vector representing the entire sentence using a NetVLAD [47] aggregation module, denoted $h(X)$. This is motivated by the recent results [34] demonstrating superior performance of NetVLAD aggregation over other common aggregation architectures such as long short-term memory (LSTM) [48] or gated recurrent units (GRU) [49].

3.3 Temporal aggregation module

Similar to input text, each input stream I_i of video descriptors is first aggregated into a single vector using temporal aggregation module h_i. For this, we use NetVLAD [47] or max pooling, depending on the input descriptors. Details are given in Section 4.

3.4 Gated embedding module

The gated embedding module $Z = f(Z_0)$ takes a d_1-dimensional feature Z_0 as input and embeds (transforms) it into a new feature Z in d_2-dimensional output space. This is achieved using the following sequence of operations:

$$Z_1 = W_1 Z_0 + b_1, \tag{1}$$

$$Z_2 = Z_1 \circ \sigma(W_2 Z_1 + b_2), \tag{2}$$

$$Z = \frac{Z_2}{\|Z_2\|_2}, \tag{3}$$

where $W_1 \in \mathbb{R}^{d_2 \times d_1}, W_2 \in \mathbb{R}^{d_2 \times d_2}, b_1 \in \mathbb{R}^{d_2}, b_2 \in \mathbb{R}^{d_2}$ are learnable parameters, σ is an element-wise sigmoid activation and $\circ$ is the element-wise multiplication

(Hadamard product). Note that the first layer, given by (1), describes a projection of the input feature Z_0 to the embedding space Z_1. The second layer, given by (2), performs context gating [34], where individual dimensions of Z_1 are reweighted using learnt gating weights $\sigma(W_2 Z_1 + b_2)$ with values between 0 and 1, where W_2 and b_2 are learnt parameters. The motivation for such gating is two-fold: (i) we wish to introduce non-linear interactions among dimensions of Z_1 and (ii) we wish to recalibrate the strengths of different activations of Z_1 through a self-gating mechanism. Finally, the last layer, given by (3), performs L2 normalization to obtain the final output Z.

3.5 Estimating text-video similarity with a mixture of embedding experts

In this section we explain how to compute the final similarity score between the input text sentence X and video Y. Recall, that each video is represented by several input streams I_i of descriptors. Our proposed model learns separate (expert) embedding between the input text and each of the input video streams. These expert embeddings are then combined together to obtain the final similarity score. More formally, we first compute a similarity score s_i between the input sentence X and input video stream I_i

$$s_i(X, I_i) = \langle f_i(h(X)), g_i(h_i(I_i)) \rangle, \tag{4}$$

where $f_i(h(X))$ is the text embedding composed of aggregation module $h()$ and gated embedding module $f_i()$; $g_i(h_i(I_i))$ is the embedding of the input video stream I_i composed of descriptor aggregation module h_i and gated embedding module g_i; and $\langle a, b \rangle$ denotes a scalar product. Please note that we learn a separate text embedding f_i for each input video stream i. In other words, we learn different embedding parameters to match the same input sentence X to different video descriptors. For example, such embedding can learn to emphasize words related to facial expressions when computing similarity score between the input sentence and the input face descriptors, or to emphasize action words when computing the similarity between the input text and input motion descriptors.

Estimating the final similarity score with a mixture of experts. The goal is to combine the similarity scores $s_i(X, I_i)$ between the input sentence X and different streams of input descriptors I_i into the final similarity score. To achieve that we employ the mixture of experts approach [46]. In detail, the final similarity score $s(X, Y)$ between the input sentence X and video Y is computed as

$$s(X, Y) = \sum_{i=1}^{N} w_i(X) s_i(X, I_i), \text{with} \quad w_i(X) = \frac{e^{h(X)^\top a_i}}{\sum_{j=1}^{N} e^{h(X)^\top a_j}}, \tag{5}$$

where $w_i(X)$ is the weight of similarity score s_i predicted from the input sentence X, $h(X)$ is the aggregated sentence representation and a_i, $i = 1 \ldots N$ the learnt parameters. Please note again that the weights w_i of experts s_i are predicted from sentence X. In other words, the input sentence provides a prior on which of the embedding experts to put more weight to compute the final global similarity score. The estimation of the weight of the different input streams can be seen as an attention mechanism that uses the input text sentence. For instance, we may expect to have high weight on the motion stream for input captions such as: "The man is practicing karate", facial descriptors for captions such as "Barack Obama is giving a talk", or on audio descriptors for input captions such as "The woman is laughing out loud".

Single text-video embedding. Please note that equation (5) can be viewed as a single text-video embedding $s(X,Y) = \langle f(X), g(Y) \rangle$, where:
$f(X) = [w_1(X) f_1(h(X)), \ldots, w_N(X) f_N(h(X))]$ is the vector concatenating individual text embedding vectors $f_i(h(X))$ weighted by estimated expert weights w_i, and $g(Y) = [g_1(h_1(I_1)), \ldots, g_N(h(I_N))]$ is the concatenation of the individual video embedding vectors $g_i(h_i(I_i))$. This is important for retrieval applications in large-scale datasets, where individual embedding vectors for text and video can be pre-computed offline and indexed for efficient search using techniques such as product quantization [50].

Handling videos with incomplete input streams. The formulation of the similarity score $s(X,Y)$ as a mixture of experts provides a proper way to handle situations where the input set of video streams is incomplete. For instance, when audio descriptors are missing for silent videos or when face descriptors are missing in shots without people. In detail, in such situations we estimate the similarity score s using the remaining available experts by renormalizing the remaining mixture weights to sum to one as

$$s(X,Y) = \sum_{i \in D} \left[\frac{w_i(X)}{\sum_{j \in D} w_j(X)} \right] s_i(X, I_i), \tag{6}$$

where $D \subset \{1 \ldots N\}$ indexes the subset of available input streams I_i for the particular input video Y. When training the model, the gradient thus only backpropagates to the available branches of both text and video.

3.6 Bi-directional ranking loss

To train the model, we use the bi-directional max-margin ranking loss [21, 22, 51, 52] as we would like to learn an embedding that works for both text-to-video and video-to-text retrieval tasks. More formally, at training time, we sample a batch of sentence-video pairs $(X_i, Y_i)_{i \in [1,B]}$ where B is the batch size. We wish to enforce that, for any given $i \in [1,B]$, the similarity score $s_{i,i} = s(X_i, Y_i)$ between video Y_i and its ground truth caption X_i is greater than every possible pair of scores $s_{i,j}$ and $s_{j,i}$, where $j \neq i$ of non-matching videos and captions. This is implemented by using the following loss for each batch of B sentence-video pairs $(X_i, Y_i)_{i \in [1,B]}$

$$l = \frac{1}{B} \sum_{i=1}^{B} \sum_{j \neq i} \Big[\max(0, m + s_{i,j} - s_{i,i}) + \max(0, m + s_{j,i} - s_{i,i}) \Big], \tag{7}$$

where $s_{i,j} = s(X_i, Y_j)$ is the similarity score of sentence X_i and video Y_j, and m is the margin. We set $m = 0.2$ in practice.

4 Experiments

In this section, we report experiments with our mixture of embedding experts (MEE) model on different text-video retrieval tasks. We perform an ablation study to highlight the benefits of our approach and compare the proposed model with current state-of-the-art methods.

4.1 Experimental setup

In the following, we describe the used datasets and details of data pre-processing and training procedures.

Datasets. We perform experiments on the following three datasets:
1 - MPII movie description/LSMDC dataset. We report results on the MPII movie description dataset [14]. This dataset contains 118,081 short video clips extracted from 202 movies. Each video has a caption, either extracted from the movie script or from transcribed audio description. The dataset is used in the Large Scale Movie Description Challenge (LSMDC). We report experiments on two LSMDC challenge tasks: movie retrieval and movie annotation. The first task evaluates text-to-video retrieval: given a sentence query, retrieve the corresponding video from 1,000 test videos. The performance is measured using recall@k (higher is better) for different values of k, or median rank (lower is better). The second, movie annotation task evaluates video-to-text retrieval: we are provided with 10,053 short clips, where each clip comes with five captions, with only one being correct. The goal is to find the correct one. The performance is measured using the accuracy. For both tasks we follow the same evaluation protocol as described on the LSMDC website[2].
2 - MSR-VTT dataset. We also report several experiments on the MSR-VTT dataset [13]. This dataset contains 10,000 unique Youtube video clips. Each of them is annotated with 20 different text captions, which results in a total of 200,000 unique video-caption pairs. Because we are only provided with URLs for each video, some of the video are, unfortunately, not available for download anymore. In total, we have successfully downloaded 7,656 videos (out of the original 10k videos). Similar to the LSMDC challenge and [14], we evaluate on the MSR-VTT dataset the text-to-video retrieval task on randomly sampled 1,000 video-caption pairs from the test set.
3 - COCO 2014 Image-Caption dataset. We also report results on the text to still image retrieval task on the 2014 version of the COCO image-caption dataset [16]. Again, we emulate the LSMDC challenge and evaluate text-to-image retrieval on randomly sampled 1000 image-caption pairs from the COCO 2014 validation set.

Data pre-processing. For text pre-processing, we use the Google News[3] trained word2vec word embeddings [53]. For sentence representation, we use NetVLAD [47] with 32 clusters. For videos, we extract frames at 25 frames per seconds and resize each frame to have a consistent height of 300 pixels. We consider up to four different descriptors

[2] `https://sites.google.com/site/describingmovies/lsmdc-2017`
[3] GoogleNews-vectors-negative300

Table 1: Ablation study on the MPII movie dataset. R@k denotes recall@k (higher is better), MR denotes mean rank (lower is better). Multiple choice is measured in accuracy (higher is better).

Evaluation task	Text-to-Video retrieval				Video-to-Text retrieval
Method	R@1	R@5	R@10	MR	Multiple Choice
MEE	**10.2**	25.0	33.1	29	74.0
MEE + Face	9.3	25.1	33.4	**27**	**74.8**
MEE + COCO	9.8	**25.6**	**34.7**	**27**	73.4
MEE + COCO + Face	10.1	**25.6**	34.6	**27**	73.9

representing the visual appearance, motion, audio and facial appearance. We pre-extract the descriptors for each input video resulting in up to four input streams of descriptors. The appearance features are extracted using the Imagenet pre-trained ResNet-152 [54] CNN. We extract 2048-dimensional features from the last global average pooling layer. The motion features are computed using a Kinetics pre-trained I3D flow network [12]. We extract the 1024-dimensional features from the last global average pooling layer. The audio features are extracted using the audio CNN [55]. Finally, for the face descriptors, we use the dlib framework[4] to detect and align faces. Facial features are then computed on the aligned faces using the same framework, which implements a ResNet CNN trained for face recognition. For each detected face, we extract 128-dimensional representation. We use max-pooling operation to aggregate appearance, motion and face descriptors over the entire video. To aggregate the audio features, we follow [34] and use a NetVLAD module with 16 clusters.

Training details. Our work was implemented using the PyTorch[5] framework. We train our models using the ADAM optimizer [56]. On the MPII dataset, we use a learning rate of 0.0001 with a batch size of 512. On the MSR-VTT dataset, we use a learning rate of 0.0004 with a batch size of 64. Each training is performed using a single GPU and takes only several minutes to finish.

4.2 Benefits of learning from heterogeneous data

The proposed embedding model is designed for learning from diverse and incomplete inputs. We demonstrate this ability on two examples. First, we show how a text-video embedding model can be learnt by augmenting captioned video data with captioned still images. For this we use the Microsoft COCO dataset [16] that contains captions provided by humans. Methods augmenting training data with still images from the COCO dataset are denoted (+COCO). Second, we show how our embedding model can incorporate an incomplete input stream of facial descriptors, where face descriptors are present in videos containing people but are absent in videos without people. Methods that incorporate face descriptors are denoted (+Face).

[4] `http://dlib.net/`

[5] `http://pytorch.org/`

Table 2: The effect of augmenting the MPII movie caption dataset with captioned still images from the MS COCO dataset. R@k denotes recall@k (higher is better), MR denotes Median Rank (lower is better) and MC denotes Multiple Choice (higher is better).

Evaluation set	COCO images				MPII videos				
Model	R@1	R@5	R@10	MR	R@1	R@5	R@10	MR	MC
MEE + Face	10.4	29.0	42.6	15	9.3	25.1	33.4	**27**	**74.8**
MEE + Face + COCO	**31.4**	**64.5**	**79.3**	**3**	**10.1**	**25.6**	**34.6**	**27**	73.9

Table 3: The effect of augmenting the MSR-VTT video caption dataset with captioned still images from the MS COCO dataset when relative image to video sampling rate $\alpha = 0.5$. R@k stands for recall@k, MR stands for Median Rank.

Evaluation set	COCO images				MSR-VTT videos			
Model	R@1	R@5	R@10	MR	R@1	R@5	R@10	MR
MEE + Face	8.4	24.9	38.9	18	13.6	37.9	51.0	10
MEE + Face + COCO	**20.7**	**54.5**	**72.0**	**5**	**14.2**	**39.2**	**53.8**	**9**

Ablation study on the MPII movie dataset. Table 1 shows a detailed ablation study on the LSMDC Text-to-Video and Video-to-Text retrieval tasks on the MPII movie dataset. Unfortunately here, we notice that incorporating heterogeneous data does not seem to significantly help the retrieval performances.

Augmenting videos with images. Next, we evaluate in detail the benefits of augmenting captioned video datasets (MSR-VTT and MPII movie) with captioned still images from the Microsoft COCO dataset. Table 2 shows the effect of adding the still image data during training. For all models, we report results on both the COCO image dataset and the MPII videos. Adding COCO images to the video training set improves performance on both COCO images but also MPII videos, showing that a single model trained from the two different data sources can improve performance on both datasets. This is an interesting result as the two datasets are quite different in terms of depicted scenes and textual captions. MS COCO dataset contains mostly Internet images of scenes containing multiple objects. MPII dataset contains video clips from movies often depicting people interacting with each other or objects.

We also evaluate the impact of augmenting MSR-VTT video caption dataset with the captioned still images from the MS COCO dataset. As the MSR-VTT is much smaller than the COCO dataset, it becomes crucial to carefully sample COCO image-caption samples when augmenting MSR-VTT during training. In detail, for each epoch, we randomly inject image-caption samples such that the ratio of image-caption samples to video-caption samples is set to a fixed sampling rate: $\alpha \in \mathbb{R}_{\geq 0}$. Note that $\alpha = 0$ means that no data augmentation is performed and $\alpha = 1.0$ means that exactly the same amount of COCO image-caption and MSR-VTT video-caption samples are used at each training epoch. Table 3 shows the effect of still image augmentation on the MSR-VTT dataset for the text-to-video retrieval task when the proportion of image-caption samples is half of the MSR-VTT video caption samples, i.e. $\alpha = 0.5$. Our proposed MEE

Table 4: Text-to-video and Video-to-Text retrieval results from the LSMDC test sets. MR stands for Median Rank, MC for Multiple Choice.

Evaluation task	Text-to-Video retrieval				Video-to-Text retrieval
Method	R@1	R@5	R@10	MR	MC Accuracy
Random baseline	0.1	0.5	1.0	500	20.0
C+LSTM+SA+FC7 [57]	4.2	13.0	19.5	90	58.1
SNUVL [52] (LSMDC16 Winner)	3.6	14.7	23.9	50	65.7
CT-SAN [2]	5.1	16.3	25.2	46	67.0
Miech *et al.* [3]	7.3	19.2	27.1	52	69.7
CCA (FV HGLMM) [20] (*same features*)	7.5	21.7	31.0	33	72.8
JSFusion [58] (LSMDC17 Winner)	9.1	21.2	34.1	36	73.5
MEE + COCO + Face (Ours)	**10.1**	**25.6**	**34.6**	**27**	**73.9**

model fully leverages the additional still images. Indeed, we observe gains in video retrieval performances for all metrics. Figure 3 shows qualitative results of our model highlighting some of the best relative improvement in retrieval ranking using the still image data augmentation. Note that many of the improved queries involve objects frequently appearing in the COCO dataset including *elephant*, *umbrella*, *baseball* or *train*.

4.3 Comparison with state-of-the-art

Table 4 compares our best approach to the state-of-the-art results on the LSMDC challenge test sets. Note that our approach significantly outperforms all other available results including JSFusion [58], which is the winning method of the LSMDC 2017 Text-to-Video and Video-to-Text retrieval challenge. We also reimplemented the normalized CCA approach from Klein *et al.* [20]. To make the comparison fair, we used our video features and word embeddings. Finally, we also significantly outperform the C+LSTM+SA+FC7 [57] baseline that augments the MPII movie dataset with COCO image caption data.

5 Conclusions

We have described a new model, called mixture of embedding experts (MEE), that learns text-video embeddings from heterogeneous data sources and is able to deal with missing video input modalities during training. We have shown that our model can be trained from image-caption and video-caption datasets treating images as a special case of videos without motion and sound. In addition, we have demonstrated that our model can optionally incorporate at training, input stream of facial descriptors, where faces are present in videos containing people but missing in videos without people. We have evaluated our model on the task of video retrieval. Our approach outperforms all reported results on the MPII Movie Description. Our work opens-up the possibility of learning text-video embedding models from large-scale weakly-supervised image and video datasets such as the Flickr 100M [59].

there is baby elephant playing with his mother
Rank: 76 → 1

people with umbrella in the rain having a confrontation with another man
Rank: 14 → 1

a baseball player falls down and the other teammates laugh at him
Rank: 26 → 2

an old women putting some liquid into a vessel to prepare something
Rank: 12 → 1

a young boy is on top of a moving train taking pictures of himself and object around him.
Rank: 23 → 1

a person is making paper art
Rank: 12 → 1

a man is eating a hot dog
Rank: 13 → 1

a boy picks up a red piece of clothing and freaks out
Rank: 29 → 1

Fig. 3: Example videos with large relative improvement in text-to-video retrieval ranking (out of 1000 test videos) on the MSR-VTT dataset when incorporating still images from the COCO dataset at training using our proposed MEE model. Notice that the improved videos involve querying objects frequently present in the COCO dataset including: *elephant*, *umbrella*, *baseball* or *train*.

Acknowledgments. The authors would like to thank Valentin Gabeur for spotting a bug in our codebase that affected multiple results from the initial paper. The bug was fixed in the following commit: `https://tinyurl.com/s6hvn9s`. This work has been partly supported by ERC grants ACTIVIA (no. 307574) and LEAP (no. 336845), CIFAR Learning in Machines & Brains program, European Regional Development Fund under the project IMPACT
(reg. no. CZ.02.1.01/0.0/0.0/15 003/0000468) and a Google Research Award.

References

1. Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., Fidler, S.: Movieqa: Understanding stories in movies through question-answering. In: CVPR. (2016)
2. Yu, Y., Ko, H., Choi, J., Kim, G.: End-to-end concept word detection for video captioning, retrieval, and question answering. In: CVPR. (2017)
3. Miech, A., Alayrac, J.B., Bojanowski, P., Laptev, I., Sivic, J.: Learning from Video and Text via Large-Scale Discriminative Clustering. In: ICCV. (2017)
4. Bojanowski, P., Lajugie, R., Grave, E., Bach, F., Laptev, I., Ponce, J., Schmid, C.: Weakly-supervised alignment of video with text. In: ICCV. (2015)
5. Alayrac, J.B., Bojanowski, P., Agrawal, N., Laptev, I., Sivic, J., Lacoste-Julien, S.: Unsupervised learning from narrated instruction videos. In: CVPR. (2016)
6. Hendricks, L.A., Wang, O., Shechtman, E., Sivic, J., Darrell, T., Russell, B.: Localizing moments in video with natural language. ICCV (2017)
7. Pan, Y., Mei, T., Yao, T., Li, H., Rui, Y.: Jointly modeling embedding and translation to bridge video and language. In: CVPR. (2016)
8. Plummer, B.A., Brown, M., Lazebnik, S.: Enhancing video summarization via vision-language embedding. In: CVPR. (2017)

9. Xu, R., Xiong, C., Chen, W., Corso, J.J.: Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In: AAAI. Volume 5. (2015) 6
10. Yu, H., Wang, J., Huang, Z., Yang, Y., Xu, W.: Video paragraph captioning using hierarchical recurrent neural networks. In: CVPR. (2016) 4584–4593
11. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: ICCV. (2015)
12. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR. (2017)
13. Xu, J., Mei, T., Yao, T., Rui, Y.: Msr-vtt: A large video description dataset for bridging video and language. In: CVPR. (2016)
14. Rohrbach, A., Rohrbach, M., Tandon, N., Schiele, B.: A dataset for movie description. In: CVPR. (2015)
15. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR, IEEE (2009) 248–255
16. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. (2014)
17. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., Bernstein, M., Fei-Fei, L.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. (2016)
18. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: ICCV, IEEE (2015) 2641–2649
19. Gong, Y., Ke, Q., Isard, M., Lazebnik, S.: A multi-view embedding space for modeling internet images, tags, and their semantics. IJCV (2014)
20. Klein, B., Lev, G., Sadeh, G., Wolf, L.: Associating neural word embeddings with deep image representations using fisher vectors. In: CVPR. (2015)
21. Wang, L., Li, Y., Lazebnik, S.: Learning deep structure-preserving image-text embeddings. In: CVPR. (2016) 5005–5013
22. Wang, L., Li, Y., Huang, J., Lazebnik, S.: Learning two-branch neural networks for image-text matching tasks. PAMI (2018)
23. Wu, C.Y., Manmatha, R., Smola, A.J., Krähenbühl, P.: Sampling matters in deep embedding learning. ICCV (2017)
24. Aytar, Y., Vondrick, C., Torralba, A.: See, hear, and read: Deep aligned representations. arXiv preprint arXiv:1706.00932 (2017)
25. Johnson, J., Karpathy, A., Fei-Fei, L.: Densecap: Fully convolutional localization networks for dense captioning. In: CVPR. (2016)
26. Pan, P., Xu, Z., Yang, Y., Wu, F., Zhuang, Y.: Hierarchical recurrent neural encoder for video representation with application to captioning. In: CVPR. (2016) 1029–1038
27. You, Q., Jin, H., Wang, Z., Fang, C., Luo, J.: Image captioning with semantic attention. In: CVPR. (2016) 4651–4659
28. Fukui, A., Park, D.H., Yang, D., Rohrbach, A., Darrell, T., Rohrbach, M.: Multimodal compact bilinear pooling for visual question answering and visual grounding. In: EMNLP. (2016) 457–468
29. Malinowski, M., Rohrbach, M., Fritz, M.: Ask your neurons: A neural-based approach to answering questions about images. In: ICCV. (2015)
30. Plummer, B.A., Mallya, A., Cervantes, C.M., Hockenmaier, J., Lazebnik, S.: Phrase localization and visual relationship detection with comprehensive image-language cues. In: ICCV. (2017)
31. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: CVPR. (2016)

32. Feichtenhofer, C., Pinz, A., Wildes, R.P.: Spatiotemporal multiplier networks for video action recognition. In: CVPR. (2017)
33. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: ICLR. (2014) 568–576
34. Miech, A., Laptev, I., Sivic, J.: Learnable pooling with context gating for video classification. arXiv preprint arXiv:1706.06905 (2017)
35. Wang, L., Xiong, Y., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: ECCV. (2016)
36. Wang, H., Schmid, C.: Action Recognition with Improved Trajectories. In: ICCV. (2013)
37. Girdhar, R., Ramanan, D., Gupta, A., Sivic, J., Russell, B.: Actionvlad: Learning spatio-temporal aggregation for action classification. In: CVPR. (2017)
38. Varol, G., Laptev, I., Schmid, C.: Long-term Temporal Convolutions for Action Recognition. PAMI (2017)
39. Arandjelovic, R., Zisserman, A.: Look, listen and learn. In: ICCV. (2017)
40. Jain, P., Netrapalli, P., Sanghavi, S.: Low-rank matrix completion using alternating minimization. In: Proceedings of the forty-fifth annual ACM symposium on Theory of computing, ACM (2013) 665–674
41. Mairal, J., Bach, F., Ponce, J., Sairo, G.: Online learning for matrix factorization and sparse coding. In: JMLR. (2010)
42. Yeh, R.A., Chen, C., Lim, T.Y., Schwing, A.G., Hasegawa-Johnson, M., Do, M.N.: Semantic image inpainting with deep generative models. In: CVPR. (2017)
43. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. arXiv preprint arXiv:1711.10925 (2017)
44. Kokkinos, I.: Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In: CVPR. (2017)
45. Tran, L., Liu, X., Zhou, J., Jin, R.: Missing modalities imputation via cascaded residual autoencoder. In: CVPR. (2017)
46. Jordan, M.I.: Hierarchical mixtures of experts and the em algorithm. Neural Computation (1994)
47. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: CVPR. (2016)
48. Hochreiter, S., Schmidhuber, J.: Long short-term memory. In: Neural Computing. (1997)
49. Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y.: On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. arXiv preprint arXiv:1409.1259 (2014)
50. Jégou, H., Douze, M., Schmid, C.: Product Quantization for Nearest Neighbor Search. PAMI **33**(1) (2011) 117–128
51. Karpathy, A., Joulin, A., Li, F.F.F.: Deep fragment embeddings for bidirectional image sentence mapping. In: NIPS. (2014)
52. Yu, Y., Ko, H., Choi, J., Kim, G.: Video captioning and retrieval models with semantic attention. In: ECCV LSMDC2016 Workshop. (2016)
53. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
54. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: CVPR. (2016)
55. Hershey, S., Chaudhuri, S., Ellis, D.P.W., Gemmeke, J.F., Jansen, A., Moore, C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B., Slaney, M., Weiss, R., Wilson, K.: CNN architectures for large-scale audio classification. In: International Conference on Acoustics, Speech and Signal Processing (ICASSP). (2017)
56. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR. (2015)
57. Torabi, A., Tandon, N., Sigal, L.: Learning language-visual embedding for movie understanding with natural-language. arXiv preprint arXiv:1609.08124 (2016)

58. Yu, Y., Kim, J., Kim, G.: A joint sequence fusion model for video question answering and retrieval. In: ECCV. (2018)
59. Thomee, B., Shamma, D.A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., Li, L.J.: Yfcc100m: the new data in multimedia research. Communications of the ACM **59**(2) (2016) 64–73