Learning Multimodal Representations for Unseen Activities

AJ Piergiovanni[1] Michael S. Ryoo[1,2]
[1]Indiana University, [2]Stony Brook University
{ajpiergi,mryoo}@indiana.edu

Abstract

We present a method to learn a joint multimodal representation space that enables recognition of unseen activities in videos. We first compare the effect of placing various constraints on the embedding space using paired text and video data. We also propose a method to improve the joint embedding space using an adversarial formulation, allowing it to benefit from unpaired *text and video data. By using unpaired text data, we show the ability to learn a representation that better captures unseen activities. In addition to testing on publicly available datasets, we introduce a new, large-scale text/video dataset. We experimentally confirm that using paired and unpaired data to learn a shared embedding space benefits three difficult tasks (i) zero-shot activity classification, (ii) unsupervised activity discovery, and (iii) unseen activity captioning, outperforming the state-of-the-arts.*

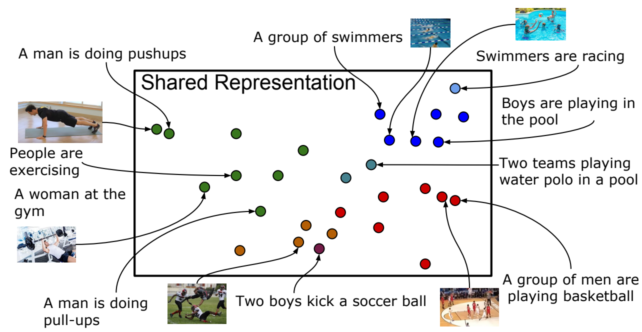


Figure 1. Taking advantage of both text and video data allows for learning of a shared representation. By utilizing unpaired text and video data, the representation naturally captures the relationships between different activities, based on the underlying relationships in word embeddings and video representations. The colors represent different activity classes of the video or sentence (e.g., various sports, pool activities, and exercises).

1. Introduction

Videos contain multiple data sources, such as visual, audio and text/caption data. Each data modality has distinct statistical properties capturing different aspects of the event. Current state-of-the-art activity recognition models [4, 42] only take visual data and class labels as input, which limits the information the model can learn from. For example, the sentence 'a group of men play basketball outdoors' contains rich information, such as 'outdoors' and 'group of men' compared to just the activity class label of 'basketball.' We desire to use such additional information to learn better representations and by doing so, we show that the learned representations are able to generalize to unseen activities (i.e., zero-shot learning).

We explore multimodal learning from video and language data, each starting with its own representation. Video data is represented as a sequence of images (spatio-temporal pixel data) while text is represented as a sequence of word embeddings (temporal data). Learning a shared representation allows for modeling the highly non-linear relationships between these modalities, capturing structure present in both video and textual data. Further, using a shared representation enables capturing similarities between concepts (e.g., basketball and volleyball both being sports with a ball) within its space by relying on either modality, even when the data is unpaired. This allows the representation to benefit from concepts not seen in both modalities during training. For example, we show taking advantage of relationships between words in pre-trained word embeddings [26] help recognize activities with no video examples. By learning a shared representation space, we transfer such relationships to video representations of potentially unseen activities. An conceptual overview of the approach is shown in Fig. 1.

Many existing approaches to both zero-shot and embedding space learning require paired data examples (e.g., examples and labeled attributes), which can be expensive to obtain. By taking advantage of adversarial learning [10], we are able effectively augment our method with **unpaired** data (i.e., random sentences and random videos without any labels or correspondence) to further improve our learned representation. By introducing many random videos and text data, we show that we are able learn representations that better capture unseen activities, without requiring any further annotations.

In this paper, we design a method capable of learning joint video/language representation using both paired and unpaired data. We experimentally confirm its benefit to three challenging tasks: (i) zero-shot activity recognition, (ii) unsupervised activity discovery, and (iii) unseen activity captioning. We show that the use of unpaired, multimodal data allows learning a shared embedding space that generalizes to unseen data.

2. Related works

Multimodal learning Previous approaches to multimodal learning have used Restricted Boltzmann Machines [41] or log-bilinear models [19] to learn distributions over sentences and images. Ngiam et al. [28] designed an autoencoder that learns joint audio-video representations, however relied on greedy, layer-by-layer training instead of training the model end-to-end. Similarly, Chandar et al. [5] proposed an autoencoder able to learn correlations between different view of images. Frome et al. [9] describe a model that maps images and words to a shared embedding. However, these works either learn a joint embedding by concatenating the different features or require a triplet consisting of positive and negative pairs; they have not explored the use/effect of unpaired data.

Text and vision Using both text and visual data has been studied for many tasks, such as image captioning [17, 15, 16] or video captioning [21, 54, 49]. Other works have explored the use of textual grounding for image/video retrieval [12, 36, 25, 14]. We note that using text for video retrieval/localization (e.g., [14]) is similar in nature to the zero-shot or unseen recognition tasks. However, in those works, there is significant overlap between the text/video examples used in training and testing, while in our work we explicitly separate the classes used during training and evaluation; we focus on 'unseen'.

There have been various models proposed to learn a fixed text embedding space with mappings from video features into this embedding space [11, 30, 39, 45, 47]. These works all learn a single directional mapping, without a shared representation space (which we find to be important). Further, most of them only learn with paired text/image samples and some require data in the form of positive/negative pairs. In this paper, we find learning a shared representation space and using unpaired, i.e., random additional data, to be important.

Learning with unpaired data Recently, there have been many works taking advantage of variational autoencoders (VAEs) [18] or generative adversarial networks (GANs) [10] to learn mappings between unpaired samples. CycleGan [55] uses a cycle-consistency loss (i.e., the ability to go from a sample in one domain to a second domain then back to the source) to learn unpaired image translation (e.g., image to sketch). Other works learn many-to-many mappings between images [2] or use two GANs to map between domains [52]. An autoencoder with shared weights for both domains has been used to learn a latent space for image-to-image translation [24]. However, these works all focus on learning mappings between unpaired data of the same modalitiy (e.g. image to image), where the data is from the same underlying distribution. We focus on a more challenging problem: learning from different modalities with very different distributions, where we find directly using previous approaches do not perform well as they are.

Zero-shot activity recognition There are works on zero-shot activity recognition. Common approaches include using attributes [23, 31, 37] or word embeddings [50, 51, 29, 38, 20] or learning a similarity metric [53, 7]. Some works have explored using adversarial losses on the latent space [6], used GANs to generate features for unseen classes [48] or used auto-encoders [46]. Felix et al. [8] proposed a GAN-based approach to learn embeddings for zero-shot learning. Different from our approach, they applied the GAN only on the semantic, hand-crafted attributes of the classes to generate representations. We formulate a more general framework generating representations for all modalities, also taking advantage of more generic and challenging text and video.

Importantly, our work differs from these previous works in three key ways: (1) we show the benefit of using additional **unpaired** samples, (2) we experimentally compare the use of the representations for three tasks (i.e., zero-shot recognition, unseen recognition, and unseen video captioning), and (3) we learn a shared, multimodal representation with bi-directional mappings in an end-to-end fashion. We find that directly using the previous methods with unpaired data do not perform as well.

3. Method

To enable learning of a shared representation, we use a deep autoencoder architecture. Our model consists of 4 neural networks:

Video Encoder $E_V : v \mapsto z_v$ **Video Decoder** $G_V : z \mapsto v$
Text Encoder $E_T : t \mapsto z_t$ **Text Decoder** $G_T : z \mapsto t$

where v is a sequence of video data and t is a sentence (sequence of words). z is the representation in the shared space that we are learning. The encoders learn a compressed representation of the video or text while the decoders are trained to reconstruct the input:

$$\mathcal{L}_{recons}(v,t) = ||G_V(E_V(v)) - v||_2 + ||G_T(E_T(t)) - t||_2 \tag{1}$$

As both text and video data are sequences, they often have different lengths. A shared representation requires that the features from both modalities have the same dimensions. Given a text representation of length L and a video representation of length T, we need to obtain a fixed-size representation. To learn a fixed-dimensional representation, there are many choices for the encoder/decoder architecture, such

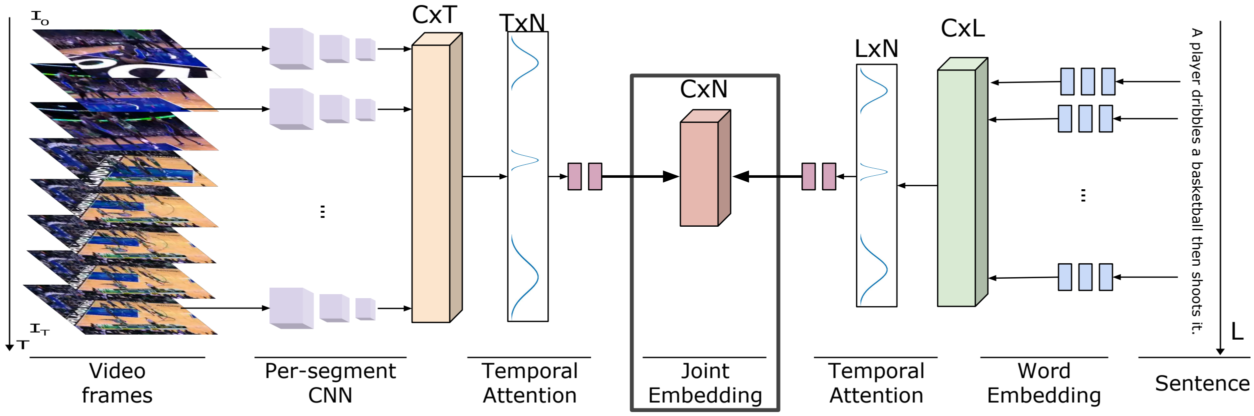


Figure 2. Illustration of the encoder models used to learn a shared representation. Videos and sentences are mapped into a low-dimensional space by applying CNNs and temporal attention. Then several fully-connected layers map to the representation. The decoders follow this same architecture with the weights transposed.

as temporal pooling [27], attention [33] or RNNs [21]. We chose temporal attention filters [33] as they learn a mapping from any length input to a N-dimensional vector and have been shown to outperform temporal pooling and RNNs on activity recognition tasks.

The attention filters consists of N Gaussians, each learning 2 parameters: a center $\hat{g}$ and width σ, which are constrained to be positive. The filters are determined by:

$$\begin{aligned} g_n &= 0.5 \cdot T \cdot (\hat{g}_n + 1) \\ F[n,t] &= \frac{1}{Z} \exp(-\frac{(t-g_n)^2}{2\sigma_n^2}) \\ & n \in \{0, 1, \dots, N-1\},\ t \in \{0, 1, \dots, T-1\} \end{aligned} \tag{2}$$

The weights are applied by matrix multiplication with the video or text sequence (e.g., the outputs of E_V or E_T): $v' = Fv$. This (i.e., v') is then used as the representations for the joint space. Additionally, we can learn a transposed version of these filters to reconstruct the input: $v = F^T v'$. To reconstruct the input, the decoders learn their own parameters with the tensors transposed, resulting in the matching output size. Fig. 2 shows our encoder architecture.

3.1. Learning a joint embedding space

To learn a joint representation space, we minimize the L_2 distance between the embeddings of a pair of text and video (shown in Fig. 3(a)):

$$\mathcal{L}_{joint}(v,t) = ||E_V(v) - E_T(t)||_2 \tag{3}$$

This forces the joint embeddings to be similar and when combined with the reconstruction loss, ensures that the representations can still reconstruct the input.

We can further constrain the networks and learned representation by forcing a cross-domain mapping from text to video and from video to text (shown in Fig. 3(b)):

$$\mathcal{L}_{cross}(v,t) = ||G_T(E_V(v)) - t||_2 + ||G_V(E_T(t)) - v||_2 \tag{4}$$

Additionally, we can use a 'cycle' loss to map from video to text and back to video. Note that while the previous losses all require paired examples, this loss does not.

$$\begin{aligned} \mathcal{L}_{cycle}(v,t) = &||G_T(E_V(G_V(E_T(t)))) - t||_2 \\ &+ ||G_V(E_T(G_T(E_V(v)))) - v||_2 \end{aligned} \tag{5}$$

To train the model to learn a joint embedding space, we minimize

$$\begin{aligned} \mathcal{L}(v,t) = &\mathcal{L}_{recons}(v,t) + \alpha_1 \mathcal{L}_{joint}(v,t) \\ &+ \alpha_2 \mathcal{L}_{cross}(v,t) + \alpha_3 \mathcal{L}_{cycle}(v,t) \end{aligned} \tag{6}$$

where α_i are hyper-parameters weighting the various loss components.

3.2. Semi-supervised learning with unpaired data

To learn using unpaired data (i.e., unrelated text and video), we use an adversarial formulation. We treat the encoders and decoders as generator networks. We then learn an additional 3 discriminator networks which constrain the generators and embedding space and force the encoders and decoders to be consistent:

(1) D_z which learns to discriminate between latent text representations and latent video representations. Conceptually, this constrains the learned embeddings to appear to be from the same distribution.

(2) D_V which learns to discriminate between true video data and generated video data $G_V(E_T(t))$.

(3) D_T which learns to discriminate between true text data and generated text data, $G_T(E_V(v))$.

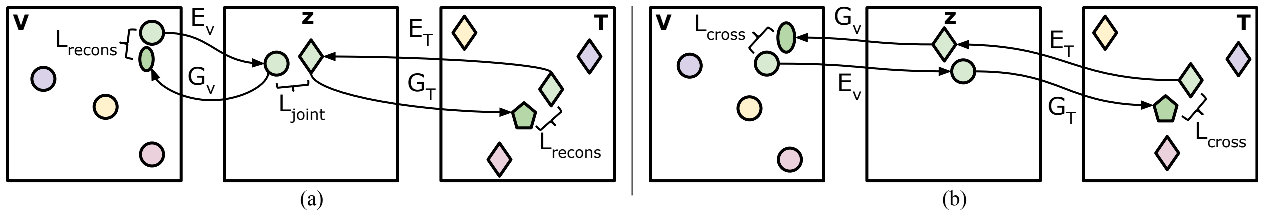


Figure 3. Visualization of several constrains on the shared embedding space. Circles are video data, ovals are reconstructed video. Diamonds are text data, and pentagons are reconstructed text. **(a)** The reconstruction (Eq. 1) and joint (Eq. 3) losses. **(b)** Mapping from text to video using the cross-domain (Eq. 4) loss.

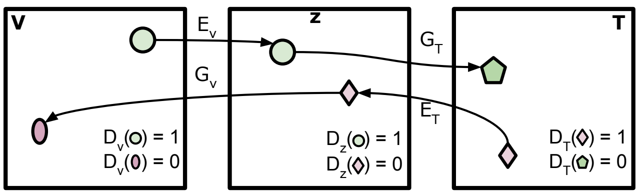


Figure 4. Visualization of the adversarial formulation to learn with unpaired data. We create 3 discriminators, (1) D_z learns to discriminate examples of text/video in the latent space. (2) D_V learns to discriminate video generated from text compared to video. (3) D_T learns to discriminate generated text compared to text.

Algorithm 1 Semi-supervised alignment with adversarial learning

function TRAIN
 for number of initialization iterations **do**
  Sample (V, T) from paired training data
  Update encoders/decoders (Eq. 6)
  Update discriminators (Eq. 7)
 end for
 for number of training iterations **do**
  Sample $P = (V_p, T_p)$ from paired and $U = (V_u, T_u)$ from unpaired training data
  Update encoders/decoders with P (Eq. 6)
  Update encoders/decoders with U (Eq. 8)
  Update discriminators based on all (Eq. 7)
 end for
end function

Given these discriminators, we minimize the following losses:

$$\begin{aligned}
\mathcal{L}_{D_z}(v,t) &= -\log(D_z(E_T(t))) - \log(1 - D_z(E_V(v))) \\
\mathcal{L}_{D_V}(v,t) &= -\log(D_V(v) - \log(1 - D_V(G_V(E_T(t)))) \\
\mathcal{L}_{D_T}(v,t) &= -\log(D_T(t)) - \log(1 - D_T(G_T(E_V(v))))
\end{aligned} \tag{7}$$

Using the discriminators, we can train the generators (encoders and decoders) to minimize the following loss based on unpaired data:

$$\begin{aligned}
\mathcal{L}_{G_z}(v,t) &= \log(D_z(E_T(t)))) + \log(1 - D_z(E_V(v))) \\
\mathcal{L}_{G_V}(v,t) &= \log(1 - D_V(G_V(E_T(t)))) \\
\mathcal{L}_{G_T}(v,t) &= \log(1 - D_T(G_T(E_V(v))))
\end{aligned} \tag{8}$$

Note that in this formulation, v and t are not paired.

These networks are trained in an adversarial setting. For example, for the text-to-video generator (i.e., $v' = G_V(E_T(t))$ and video discriminator, D_V, we optimize the following minimax equation:

$$\begin{aligned}
\min_{E_T,G_V} \max_{D_V} &= \mathbb{E}_{v\sim p_{\text{data}}(v)}[\log D_V(v)] \\
&+ \mathbb{E}_{t\sim p_{\text{data}}(t)}[\log(1 - D_V(G_V(E_T(t))))]
\end{aligned} \tag{9}$$

This equation is similarly applied for video-to-text. For learning the embedding space with the video and text encoders, E_V, E_T and the discriminator D_z, we optimize the following minimax equation:

$$\begin{aligned}
\min_{E_T,E_V} \max_{D_z} &= \mathbb{E}_{v\sim p_{\text{data}}(v)}[\log D_z(E_V(v))] \\
&+ \mathbb{E}_{t\sim p_{\text{data}}(t)}[\log(1 - D_z(E_T(t)))]
\end{aligned} \tag{10}$$

As training GANs can be unstable, we developed a method to allow for more stable training of the joint embedding space, shown in Algorithm 1. We initialize both the generator and discriminator networks by training only on paired data. After several iterations of this, we train with both unpaired and paired data. We found the initial training of the generators and discriminators was important for stability, without it the loss often diverges and the learned embedding did not generalize to unseen activities.

4. Experiments

We compare our various approaches on different tasks (i) zero-shot activity recognition, (ii) unsupervised activity discovery and (iii) unseen activity captioning. These tasks test various combinations of our encoders and decoders and how well the shared representation generalizes to unseen data. We experimentally confirm the benefits of our methods using multiple public datasets: AcitivtyNet [13, 21], HMDB [22], UCF101 [40], and MLB-YouTube [34]. Implementation details can be found in the Appendix.

Table 1. Comparison of accuracy of various methods on ActivityNet for 5, 10, 20 or 50 unseen classes. These results are averaged over 10 trials where each trial has a different set of unseen activities.

	5 Unseen	10 Unseen	20 Unseen	50 Unseen
Paired Data				
Fixed Text Representation	41.9	38.4	29.4	15.6
Triplet Loss	56.8	44.9	38.8	23.3
joint	54.3	41.7	36.1	21.2
recons + cross	21.1	12.6	7.6	2.9
joint + recons	70.1	54.4	42.6	27.5
joint + recons + cycle	70.4	54.3	42.1	26.8
joint + recons + cross	72.6	55.4	43.2	27.8
joint + recons + cross + cycle	76.4	56.9	45.5	28.8
triplet + recons + cross + cycle	76.7	57.2	46.3	29.1
With Adversarial Losses (triplet + recons + cross + cycle + Adv.)				
+ D_z	78.5	57.4	45.9	29.3
+ $D_v + D_t$	77.4	57.2	45.7	28.9
+ $D_z + D_v + D_t$	79.8	58.4	46.5	29.8
Paired + Unpaired Data				
recons + cycle	22.8	13.6	8.4	4.2
triplet + recons + cycle	72.6	58.4	44.7	29.3
triplet + recons + cross + cycle	73.4	59.1	45.3	29.2
Without Algorithm 1	23.4	11.7	6.5	3.1
All terms	82.5	60.4	46.2	30.1

Baselines For baselines, we compare to a fixed-text embedding space, were only a mapping from video data into the text embedding space is learned (e.g., [30]). We also compare to learning a shared embedding space with the 'recons' (Eq. 1) and 'cross' (Eq. 4) terms (e.g., [28]). We additionally compare to methods like CycleGan [55], using various components without Algorithm 1.

4.1. Zero-shot activity recognition

Zero-shot activity recognition is the problem of classifying a video that belongs to a class not seen during training. Given training videos of seen classes together with paired text descriptions, our approach learns a shared embedding that maps videos/texts from multiple seen classes. The objective is to classify videos of unseen classes solely based on the learned embedding space and the text samples.

To enable recognition of unseen activities, we use a sentence of the new, unseen class and obtain its representation in the shared space. We can then obtain representations of videos in the same space, using nearest neighbors matching to classify each clip. Such approach takes advantage of the learned textual relationships (e.g., [26]) and the shared, multimodal representation space.

We use the ActivityNet captions [21] dataset to learn the shared representations, as this dataset has both sentence descriptions for each video as well as activity classes. We randomly choose a set of K activity classes and withhold all videos/sentences belonging to those classes during training. For evaluating on the unseen activities, we take a subset of sentences for the unseen classes and map the sentences into the joint embedding space, $z_t = E_T(t)$. We then map the videos into the space, $z_v = E_V(v)$ and use nearest neighbors to match each video (z_v) to text (z_t), using the class of the nearest sentence as the classification for the video. We rely on the similarities between the representations (e.g., word embeddings) to enable the models ability to generalize to these unseen classes.

In Table 1, we compare the effect of the various loss components. For each method, we run 10 trials each with a different set of unseen activity classes and average the results. We find that previous methods of learning a fixed language embedding (e.g., [38, 50, 51]) are significantly outperformed by learning a joint representation. Previous methods learning embedding spaces without the 'joint' term (e.g. [28]), we found yield nearly random performance on these tasks, suggesting that forcing the representations to match in the embedding space is important. Further, adding the reconstruction, cross-domain, and cycle losses all improve performance. We also compare to a standard triplet loss (e.g., [11]) which requires positive/negative samples. We find that the triplet loss outperforms the 'joint' loss, but is surpassed by adding the 'cycle' and 'cross' terms, which use less data. We also compared using the triplet loss when combined with the other terms, finding a slight improvement over the joint term. Note using both the joint and triplet would be redundant, since the triplet loss contains the joint loss terms.

We also compare the various components of the adver-

Table 2. Results on HMDB51 and UCF101 (accuracy) compared to previous state-of-the-art results. We find that learning a shared representation is beneficial and that augmented with unpaired data provides the best results.

	Feat	HMDB51	UCF101
SJE [1]	IDT	12.0 ± 2.6	9.3 ± 1.7
ConSe [29]	IDT	15.0 ± 2.7	11.6 ± 2.1
ZSECOC [35]	IDT	22.6 ± 1.2	15.1 ± 1.7
SE [50]	IDT	21.2 ± 3.0	18.6 ± 2.2
MRR [51]	IDT	24.1 ± 3.8	22.1 ± 2.5
SAE [20]	I3D	25.6 ± 3.2	25.4 ± 2.2
Ours (paired)	IDT	26.3 ± 3.2	25.4 ± 3.4
Ours (paired + unpaired)	IDT	29.7 ± 2.2	26.4 ± 2.1
Ours (paired)	I3D	28.3 ± 2.7	27.8 ± 2.2
Ours (paired + unpaired)	I3D	34.7 ± 2.4	33.4 ± 1.8

Table 3. Comparison of various source of unpaired data on ActivityNet with 10 unseen classes, values reported for both unseen classes and all (seen+unseen) classes. Results are accuracy, higher is better.

	Unseen	All
Paired Data	58.3	69.6
+ Random Wikipedia Sentences	55.8	66.4
+ Random Dictionary Defs.	56.3	68.2
+ Verb Dictionary Defs.	59.2	70.7
+ Random YouTube Videos	58.7	70.1
+ Verbs + Random Videos	60.3	71.2

sarial loss. We compare to having just the adversarial loss on the representation (D_z), like [6], and compare just the adversary on the generated videos/sentences. We find the use of all terms is important for performance.

While previous works such as [28] can support learning with unpaired data, we find that the adversarial loss provides better results than just the 'cycle' and 'recons' terms, and further improves over training with just paired data. Further, we find that CycleGan-style approaches, without Algorithm 1, fail on this task.

In Table 2, we compare our approach to previous zero-shot learning methods on HMDB and UCF101. The paired training data for these models is drawn from ActivityNet with any classes belonging to HMDB or UCF101 withheld. The unpaired text data is sampled from Charades and the video data comes from either HMDB (when testing on UCF101) or UCF101 (when testing on HMDB). As HMDB and UCF101 have no text descriptions, we created a sentence description for each activity class (included in Appendix B). We find that the shared representation outperforms the previous approaches on these datasets and unpaired adversarial learning further improves performance.

4.2. Use of Unpaired Data

We explore different strategies for obtaining unpaired data. Keeping a fixed set of paired text and videos, we explore adding various sources of unpaired data: (i) 10k random Wikipedia sentences, (ii) 10k random dictionary definitions, and (iii) 10k verb dictionary definitions. We also compare adding 10k random videos from YouTube as additional video data. Ours results using 10 unseen classes are in Table 3. We find that augmenting with similar unpaired data improves performance, while irrelevant data harms performance. We find that dictionary verb definitions improve performance the most, as they capture important semantic information regarding the activities we are learning. The use of additional video data is further beneficial.

Table 4. Comparison of unsupervised activity classification on MLB-YouTube.

	Accuracy	mAP
Baseline I3D features	23.4	32.6
Fixed Text Representation	27.9	34.7
joint	34.5	41.6
joint + recons	37.9	43.7
joint + recons + cycle	44.2	48.6
joint + recons + cross	43.7	49.3
triplet + recons + cross	43.9	49.5
All (paired)	48.4	51.2
All (+ unrelated unpaired)	39.7	43.9
All (+ related unpaired)	49.1	54.3

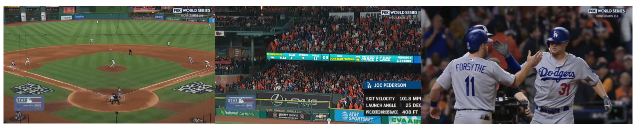
4.3. Unsupervised activity discovery

To further evaluate the shared representation, we conducted experiments on unsupervised activity discovery. For this task, we expanded the MLB-YouTube dataset [34] by densely annotating the videos with a transcription of the announcers' commentary, resulting in approximately 50 hours of aligned text and video. Examples of this data are shown in Fig. 5. The MLB-YouTube dataset is designed for fine-grained activity recognition, where the difference between activities is quite small. Additionally, these captions only roughly describe what is happening in the video, and often contain unrelated stories or commentary on a previous event, making this a challenging task. The dataset will be made publicly available. To train the shared representation, we split each baseball video into 30 second intervals and use the corresponding text as paired data, resulting in 6,089 paired training samples.

We evaluate the shared representation using the segmented videos from MLB-YouTube. For each video, we compute the embedded features and apply k-means clustering ($k = 8$, the number of classes). Each segmented video is assigned to a cluster and votes for the cluster label based on its ground truth label. We use that cluster assignment for classification on the MLB-YouTube test set. We report our findings in Table 4. As a baseline, we cluster I3D features pre-trained on Kinetics. We find that our methods improve the representation. However, we note that when using unpaired data from Charades, the performance drops. This is likely due to Charades data being very different from MLB-YouTube data. We collected additional captions and baseball videos to augment the MLB-YouTube dataset, and confirmed that unpaired data helps when it is from a similar

Table 5. Unseen activity recognition results (accuracy) on ActivityNet, HMDB51 and UCF101, evaluated by using both unseen and seen classes for the testing.

	ActNet (10 unseen)	ActNet (50 unseen)	HMDB51	UCF101
Fixed Text Representation	55.7	46.8	24.5	26.8
Triplet Loss	57.7	48.5	27.6	29.8
joint	62.1	50.2	29.8	30.6
joint + recons	64.4	52.6	30.4	31.3
joint + recons + cross + cycle	69.6	58.5	35.6	36.5
triplet + recons + cross + cycle	69.8	58.6	35.7	36.8
Paired + Unpaired Data				
All terms	71.7	65.9	38.9	42.2



He got right on top of that pitch, Pederson, and shot and way out of here. Three-run blast.

That has been a feat in this series for both teams, nobody is hitting with two strikes. That's how good the pitching has been.

They would suspend him at the beginning of next year as opposed to for a game during this World Series.

He is an aggressive third baseman and he can really play over there you know. He definitely takes pride in his defense as well.

Figure 5. Example video sequences from the MLB-YouTube dataset with the commentary caption. **Top**: Sentences that describe the occurring activities. **Bottom**: Sentences that do not describe the current activities.

Table 6. Comparison of unsupervised activity classification on HMDB and UCF101.

	HMDB	UCF101
I3D features	26.6	42.5
Joint	32.4	57.7
Joint + recons	33.5	59.0
All (paired)	34.6	59.5
All (+ unpaired)	34.9	59.9

distribution.

In Table 6 we compare various methods for unsupervised activity discovery on HMDB and UCF101. Here, we learn a shared representation using the ActivityNet videos and captions. We withhold any videos belonging to a class in HMDB or UCF101. Unlike MLB-YouTube, on these datasets, we find that using the unpaired training with Charades further improves performance. This confirms that when the additional data is similar to the target dataset, using the adversarial learning setting further improves the representations.

4.4. Unseen video captioning

As our model learns a bi-directional mappings, we can apply our model to generate video captions. Existing video captioning models are unable to create realistic captions for unseen activities, as without training data they do not know the words to describe the video. Given a video, v, we can generate a caption by mapping the video to text $t = G_T(E_V(v))$. For each word, we then use nearest neighbors matching with the GloVe embeddings to obtain the words to form a sentence. We find that using our method with paired and unpaired data improves performance using METEOR (3.6 to 6.9) [3] and CIDEr [44] (8.9 to 13.9) scores. For these metrics, higher values are better and are measured with the unseen classes from the ActivityNet dataset. In Table 8, we report the commonly used METEOR [3] and CIDEr [44] scores of our various models, measured with the unseen classes from the ActivityNet dataset. We find that learning a shared representation (4.1) is beneficial and using unpaired samples further improves the task (5.3 paired only vs 6.9 paired and unpaired). In Fig. 6, we show example captioned videos. Note that this task is extremely challenging, as it requires the model to generate captions using activity words (e.g., basketball) not seen during training.

5. Conclusion

We proposed an approach to learn a joint language/text representation using various constraints. We further extended the model to be able to learn with unpaired video and text data using an adversarial formulation. We experimentally confirmed that learning with unpaired data is beneficial to three difficult tasks (i) zero-shot activity classification, (ii) unsupervised activity discovery, and (iii) unseen activity

Several men are playing basketball   People are swimming in the ocean

Figure 6. Example captions for unseen activities. **Left**: Using a shared representation allows the model to correctly caption this video as basketball, despite never seeing an example of basketball during training. **Right**: An example of a caption for the unseen water-ski activity. Here the model fails to correctly caption the activity.

Table 7. Comparison of several models for unseen activity captioning using the ActivityNet dataset, using METEOR and CIDEr scores. This evaluation was done on 10 unseen classes held out during training. Higher values are better.

	METEOR	CIDEr
Fixed Text Representation	3.64	8.95
Joint	4.21	9.23
All (paired)	5.31	11.21
All (paired + unpaired)	6.89	13.95

captioning. We find that the use of related unpaired data is beneficial. We presented several strategies for obtaining unpaired data and confirmed the benefit of adding additional, relevant unpaired data.

Acknowledgement This work was supported in part by the National Science Foundation (IIS-1812943 and CNS-1814985).

References

[1] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[2] A. Almahairi, S. Rajeswar, A. Sordoni, P. Bachman, and A. Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data. *arXiv preprint arXiv:1802.10151*, 2018.

[3] S. Banerjee and A. Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005.

[4] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] S. Chandar, M. M. Khapra, H. Larochelle, and B. Ravindran. Correlational neural networks. *Neural computation*, 28(2):257–285, 2016.

[6] L. Chen, H. Zhang, J. Xiao, W. Liu, and S.-F. Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[7] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2005.

[8] R. Felix, V. B. Kumar, I. Reid, and G. Carneiro. Multi-modal cycle-consistent generalized zero-shot learning. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 21–37, 2018.

[9] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[11] S. Guadarrama, N. Krishnamoorthy, G. Malkarnenkar, S. Venugopalan, R. Mooney, T. Darrell, and K. Saenko. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2013.

[12] S. Gupta and R. J. Mooney. Using closed captions as supervision for video activity recognition. In *Proceedings of the American Association for Artificial Intelligence (AAAI)*, 2010.

[13] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015.

[14] L. Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, and B. Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[15] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[16] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[17] A. Karpathy, A. Joulin, and L. F. Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[18] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *International Conference on Learning Representations (ICLR)*, 2014.

[19] R. Kiros, R. Salakhutdinov, and R. Zemel. Multimodal neural language models. In *International Conference on Machine Learning (ICML)*, 2014.

[20] E. Kodirov, T. Xiang, and S. Gong. Semantic autoencoder for zero-shot learning. *arXiv preprint arXiv:1704.08345*, 2017.

[21] R. Krishna, K. Hata, F. Ren, L. Fei-Fei, and J. C. Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.

[23] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2011.

[24] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[25] A. Miech, J.-B. Alayrac, P. Bojanowski, I. Laptev, and J. Sivic. Learning from video and text via large-scale discriminative clustering. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[27] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[28] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *International Conference on Machine Learning (ICML)*, 2011.

[29] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013.

[30] M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, and N. Yokoya. Learning joint representations of videos and sentences with web image search. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

[31] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell. Zero-shot learning with semantic output codes. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[32] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[33] A. Piergiovanni, C. Fan, and M. S. Ryoo. Learning latent sub-events in activity videos using temporal attention filters. In *Proceedings of the American Association for Artificial Intelligence (AAAI)*, 2017.

[34] A. Piergiovanni and M. S. Ryoo. Fine-grained activity recognition in baseball videos. In *CVPR Workshop on Computer Vision in Sports*, 2018.

[35] J. Qin, L. Liu, L. Shao, F. Shen, B. Ni, J. Chen, and Y. Wang. Zero-shot action recognition with error-correcting output codes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[36] A. Rohrbach, M. Rohrbach, R. Hu, T. Darrell, and B. Schiele. Grounding of textual phrases in images by reconstruction. In *Proceedings of European Conference on Computer Vision (ECCV)*. Springer, 2016.

[37] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning (ICML)*, 2015.

[38] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[39] Y. C. Song, I. Naim, A. Al Mamun, K. Kulkarni, P. Singla, J. Luo, D. Gildea, and H. A. Kautz. Unsupervised alignment of actions in video with text descriptions. In *IJCAI*, pages 2025–2031, 2016.

[40] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[41] N. Srivastava and R. R. Salakhutdinov. Multimodal learning with deep boltzmann machines. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.

[42] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. *arXiv preprint arXiv:1711.11248*, 2017.

[43] M. Tygert, A. Szlam, S. Chintala, M. Ranzato, Y. Tian, and W. Zaremba. Convolutional networks and learning invariant to homogeneous multiplicative scalings. *arXiv preprint arXiv:1506.08230*, 2015.

[44] R. Vedantam, C. Lawrence Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[45] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[46] W. Wang, Y. Pu, V. K. Verma, K. Fan, Y. Zhang, C. Chen, P. Rai, and L. Carin. Zero-shot learning via class-conditioned deep generative models. *arXiv preprint arXiv:1711.05820*, 2017.

[47] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[48] Y. Xian, T. Lorenz, B. Schiele, and Z. Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[49] H. Xu, B. Li, V. Ramanishka, L. Sigal, and K. Saenko. Joint event detection and description in continuous video streams. *arXiv preprint arXiv:1802.10250*, 2018.

[50] X. Xu, T. Hospedales, and S. Gong. Semantic embedding space for zero-shot action recognition. In *International Conference on Image Processing (ICIP)*, 2015.

[51] X. Xu, T. Hospedales, and S. Gong. Transductive zero-shot action recognition by word-vector embedding. *International Journal of Computer Vision (IJCV)*, 2017.

[52] Z. Yi, H. Zhang, P. Tan, and M. Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[53] Z. Zhang and V. Saligrama. Zero-shot learning via semantic similarity embedding. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[54] L. Zhou, Y. Zhou, J. J. Corso, R. Socher, and C. Xiong. End-to-end dense video captioning with masked transformer. *arXiv preprint arXiv:1804.00819*, 2018.

[55] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Table 8. Comparison of several models for unseen activity captioning using the ActivityNet dataset, using METEOR and CIDEr scores. This evaluation was done on 10 unseen classes held out during training. Higher values are better.

	METEOR	CIDEr
Fixed Text Representation	3.64	8.95
Joint	4.21	9.23
All (paired)	5.31	11.21
All (paired + unpaired)	6.89	13.95

A. Implementation/training details

We implement our models in PyTorch. For the per-segment video CNN, we use I3D [4] to obtain a $1024 \times T$ video representation. We trained a version of I3D based on Kinetics-600, but withheld all classes that appear in ActivityNet, HMDB51, or UCF101 so that the classes are truly unseen. This resulted in a training set with 478 classes and 278k videos. Since generating videos is an extremely challenging task, the video autoencoders start with and generate the I3D feature. We use GloVe word embeddings [32] to obtain a language representation. We set $N = 4$ for the temporal attention filters and apply 4 fully connected layers. These layers are followed by L_2 normalization so that the embedding space has unit length [43]. We train the models for 200 epochs and use stochastic gradient descent with momentum to minimize the loss function with a learning rate of 0.01. After every 50 epochs, we decay the learning rate by a factor of 10. When training in the adversarial setting (e.g., Algorithm 1 in the main paper), we initialize the network training for 50 epochs on paired data followed by 200 on the paired + unpaired data.

A.1. Unseen video captioning

As our model learns a bi-directional mappings, we can apply our model to generate video captions. Existing video captioning models are unable to create realistic captions for unseen activities, as without training data they do not know the words to describe the video. Given a video, v, we can generate a caption by mapping the video to text $t = G_T(E_V(v))$. For each word, we then use nearest neighbors matching with the GloVe embeddings to obtain the words to form a sentence. In Table 8, we report the commonly used METEOR [3] and CIDEr [44] scores of our various models, measured with the unseen classes from the ActivityNet dataset. We find that learning a joint representation is beneficial and using unpaired samples further improves the task. Note that this task is extremely challenging, as it requires the model to generate captions using activity words (e.g., basketball) not seen during training.

B. Additional Experiments

B.1. Comparison of temporal pooling methods

To confirm that temporal attention is beneficial, we compare different forms of temporal pooling (i) max-pooling, (ii) sum-pooling, (iii) LSTM, and (iv) temporal attention filters [33]. In Table 9, we compare these temporal pooling methods learning the joint embedding space. We confirm that using the temporal attention filters performs best.

Table 9. Comparison of temporal pooling methods for 5 unseen classes in the ActivityNet dataset.

	Accuracy
Max Pooling	23.4
Sum Pooling	24.1
LSTM	42.3
Temporal Attention Filters	55.2

Table 10. Comparison of different ratios of paired and unpaired data methods for 5 unseen classes in the ActivityNet dataset.

Paired/Unpaired	Accuracy
100% / 0%	74.2
75% / 25%	73.2
50% / 50%	69.7
25% / 75%	62.6
0% / 100%	24.5

B.2. Comparison of different ratios of paired and unpaired data

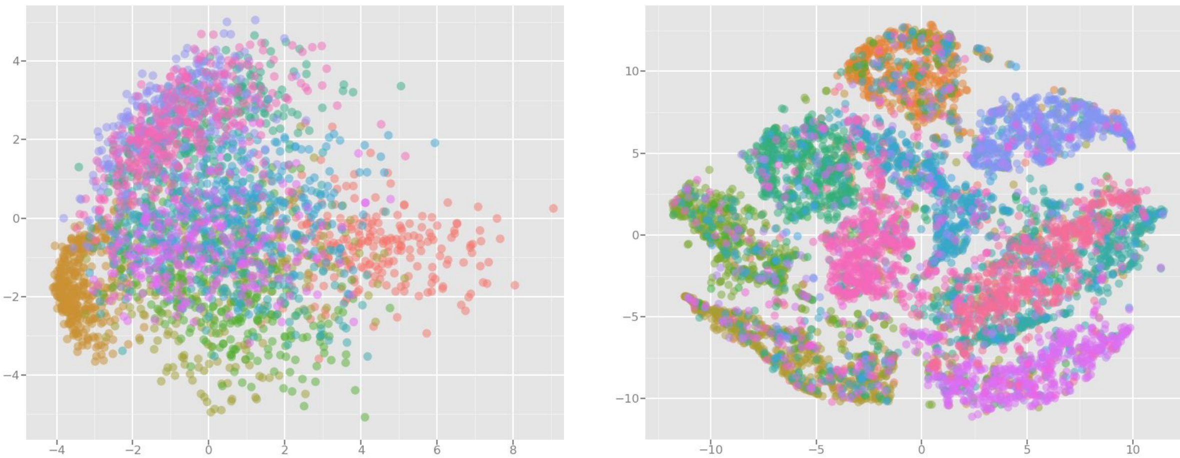
We compare different ratios of paired and unpaired data to see how much paired data we require and how much unpaired data is beneficial. For these experiments, we use all the loss terms (i.e., what provided us the best results). Note that in these experiments, the total number of samples was the same for each method (40k examples) so that we can directly compare the effects of unpaired data vs. paired data. Thus not all the available data was used.

In Table 10, we show the results. We find that using no paired data results in nearly random performance, but using using some paired data greatly improves the embedding space. The model using 100% paired data performs best, as all the others are using less overall paired data.

We also compare augmenting our 40k paired training samples with different amounts of unpaired data. Since UCF101 and HMDB only have 13k and 7k examples, to get up to 60k samples, we also use videos from the Kinetics dataset [4]. The results, shown in Table 11, show that adding the initial 10k samples is most beneficial, while additional samples do not seem to meaningfully improve results. However, due to our training method where each batch consists of 50% paired data and 50% unpaired data, the additional unpaired data does not harm results either.

B.3. MLB-Youtube Captions

In Fig. 7, we compare t-SNE embeddings of the fixed text representation and our joint embedding space. This visually shows that learning a joint embedding space gives more distinct class distributions.



(a) (b)

Figure 7. t-SNE mapping of **(a)** fixed text representation and **(b)** joint embedding with all paired losses for the MLB-YouTube dataset. The joint embedding space provides most distinct representations for the activities. Each color represents the activity class of the video (e.g., swing, hit, foul ball, etc.).

Table 11. Comparison using 40k paired examples and varying amounts of unpaired samples for 5 unseen classes in the ActivityNet dataset.

Unpaired Samples	Accuracy
0	77.1
10k	82.4
20k	83.9
40k	83.6
60k	83.5

Table 12. Comparison of several models for standard, seen video captioning using the MLB-YouTube dataset, using Bleu, METEOR and CIDEr scores. Higher values are better.

	Bleu	METEOR	CIDEr
Fixed Text Representation	0.12	0.04	0.12
Joint Representation	0.14	0.08	0.15
Joint + all paired	0.15	0.10	0.18
Joint + paired + unpaired	0.10	0.02	0.08

B.3.1 MLB-YouTube Captions

As a baseline for the MLB-YouTube captions dataset, we compared several different models for standard video captioning (i.e., all activity classes are seen). This task is quite challenging compared to other datasets as the announcers commentary is not always a direct description of the current events. Often the announcers tell loosely related stories and attempt to describe events differently each time to avoid repetition. Additionally, the descriptions contain on average 150 words for each 30 second interval and current captioning approaches usually only trained and tested on 10-20 word sentences. Due to these factors, this task is quite challenging the standard evaluation metrics do not account for these factors. In Table 12, we report our results on this task.

Table 13. Comparison of various pronouns on the UCF101 dataset with 50 unseen classes.

	Accuracy
Baseline Sentences	33.4
All ‘man’	33.2
All ‘woman’	33.3
All ‘person’	33.4
Random pronoun	33.4

C. HMDB and UCF101 Sentences

For the HMBD and UCF101 datasets, we created sentences to describe each activity class. Our sentences descriptions are included in this appendix.

These sentences are written for each activity class (by randomly selecting a single video per class) and are shared for all instances of the activity. Depending on what video was randomly chosen for the class, some sentences describe the actor as a ‘man’, ‘woman’, or ‘person’ which could confuse the model. Ideally, the CNN embedding needs to learn to ignore the impact of such pronoun changes.

We conducted experiments comparing randomly replacing the pronouns to determine if there was any bias introduced by the pronouns. We show the results in Table 13. We find that the choice of pronouns does not impact performance, as our model automatically learns to focus more on verbs rather than pronouns. When examining the temporal attention filters on the sentences, we found that they placed very little ‘attention’ on the start of the sentence, where the pronoun usually is, suggesting that the pronoun has very little effect on the embedding space we learned.

HMDB:

1. chew: a woman is chewing on bread

2. golf: a man swings a golf club
3. sword exercise: a person is playing with a sword
4. walk: a person is walking
5. jump: a person jumps into the water
6. pour: a man pours from a bottle
7. laugh: a man is laughing
8. shoot gun: a person rapidly fires a gun
9. run: a person is running
10. turn: a person turns around
11. ride bike: a man is riding a bike on the street
12. swing baseball: a boy hits a baseball
13. draw sword: a person draws a sword
14. sit: a person sits in a char
15. fencing: two men are fencing
16. dribble: a boy dribbles a basketball
17. stand: a person stands up
18. pushup: a man does pushups
19. sword: two people are fighting with swords
20. pullup: a boy does pullups in a doorway
21. smile: a man smiles
22. shake hands: two people shake hands
23. shoot ball: a person shoots a basketball
24. kick: a person kicks another person
25. somersault: a person does a somersault
26. flic flac: a boy does a backflip
27. hug: two people hug
28. hit: a boy swings a baseball bat
29. dive: a person jumps into a lake
30. drink: a man drinks from a bottle
31. punch: a woman punches a man
32. wave: a person waves their hand
33. talk: a person is talking
34. kiss: a man and woman kiss
35. catch: a boy catches a ball
36. smoking: a woman smokes a cigarette
37. eat: a man eats pizza
38. throw: a person throws a ball
39. climb stairs: a man is running down the stairs
40. kick ball: a person kicks a soccer ball
41. ride horse: a girl is riding a horse
42. fall floor: a man is pushed onto the ground
43. brush hair: a girl is brushing her hair
44. situp: a man does situps
45. cartwheel: a guy runs and jumps and flips
46. pick: a man picks a book
47. push: a boy pushes a table
48. climb: a man is climbing up a wall
49. handstand: three girls do handstands
50. clap: a woman claps her hands
51. shoot bow: a person shows a bow and arrow

UCF101:

1. MilitaryParade: people are marching and waving a flag
2. TrampolineJumping: kids are jumping on a trampoline
3. PlayingDaf: a person moves a circle and hits it
4. SalsaSpin: poeple are dancing and spinning
5. CuttingInKitchen: a person is in the kitchen using a knife
6. ApplyEyeMakeup: a woman is putting on makeup
7. PlayingViolin: a person plays the violin
8. YoYo: a person plays with a yoyo
9. PlayingCello: a person is playing the cello
10. Bowling: a person is bowling
11. UnevenBars: a woman is spinning and flying on bars
12. BalanceBeam: a woman is on the balance beam
13. SkyDiving: people are falling out of the sky
14. SumoWrestling: two fat people are wrestling
15. PushUps: a man does pushups

16. FloorGymnastics: a girl does gymnastics
17. ApplyLipstick: a woman is putting on lipstick
18. BreastStroke: a woman is swimming
19. GolfSwing: a man swings a golf club
20. PlayingDohl: a person hits on a drum
21. HorseRiding: a woman rides a horse
22. PlayingFlute: a person blow into a flute
23. PizzaTossing: a man is making a pizza
24. CleanAndJerk: a person is lifting weights
25. WritingOnBoard: a person is writing on the wall
26. CricketShot: a person hits a ball with a bat
27. FieldHockeyPenalty: a girl in the field shoots a ball
28. HammerThrow: a person spins and throws an object
29. BodyWeightSquats: a man is squatting
30. CliffDiving: a person jumps off a cliff
31. Typing: a person is typing at a computer
32. MoppingFloor: a man mops the floor
33. TaiChi: people are doing tai chi
34. PlayingPiano: a person plays piano
35. Punch: someone punches another person
36. Nunchucks: a person swings nun chucks
37. RopeClimbing: a person climbs a rope
38. Swing: a baby is swinging
39. Knitting: a woman is knitting
40. Rafting: people are rafting on a river
41. PlayingGuitar: a person strums a guitar
42. ShavingBeard: a man shaves his beard
43. JugglingBalls: a person is juggling balls
44. Diving: a boy dives into a pool
45. JumpingJack: a person jumps and swings his arms
46. VolleyBallSpiking: people hit a volleyball
47. PoleValut: a person runs with a pole and launches into the air
48. SkateBoarding: a man is skateboarding
49. BoxingPunchingBag: a man is punching a bag
50. IceDancing: people are ice skating
51. WallPushups: a person does pushups against a wall
52. FrisbeeCatch: a person jumps and catches a frisbee
53. Drumming: people are drumming
54. JumpRope: a girl is jumping rope
55. HeadMassage: a person gets their head massaged
56. PlayingTabla: a person plays two drums
57. TableTennisShot: people are playing table tennis
58. PommelHorse: a person spins around on their hands
59. HighJump: a man jumps over a bar and lands on his back
60. BasketballDunk: a man jumps and dunks the basketball
61. BoxingSpeedBag: a man punches a bad in the air quickly
62. PullUps: a person does hangs on a bar and pulls up
63. RockClimbingIndoor: a person is climbing up rocks
64. BlowingCandles: a boy blows out candles on a cake
65. Skiing: people are skiing on a mountain
66. WalkingWithDog: a person walks a dog
67. Basketball: men are playing basketball
68. SoccerJuggling: a person is playing with a soccer ball
69. Fencing: people are fencing
70. Billiards: a man is playing billiards
71. BaseballPitch: a man throws a baseball
72. BlowDryHair: a woman is drying her hair
73. CricketBowling: a person throws a cricket ball
74. BandMarching: people are walking down the street playing music
75. PlayingSitar: a person plays a funny guitar
76. ThrowDiscus: a person spins and throws a disk
77. StillRings: a man holds in the air on rings
78. Lunges: a person bends to the ground with one knee
79. Skijet: a person rides a jetski in the ocean

80. BabyCrawling: a baby is crawling on the floor
81. Mixing: a woman is mixing in a bowl
82. Hammering: a person is hitting nails with a hammer
83. Shotput: a person spins and launches a ball
84. Archery: a man shoots a bow and arrow
85. Surfing: a man is surfing in the ocean
86. FrontCrawl: a person is swimming freestyle
87. HulaHoop: a person spins a hoop around their waist
88. JavelinThrow: a person throws a spear
89. Rowing: people are in a canoe and rowing
90. Kayaking: a person is kayaking on a lake
91. ParallelBars: a man does gymnastics on the parallel bars
92. HorseRace: horses are racing around a track
93. HandstandWalking: a person stands on their hands and walk
94. BrushingTeeth: a boy brushes his teeth
95. LongJump: a person runs and jumps into a sand pit
96. Biking: people are riding bikes
97. HandstandPushups: a person does pushups upside down
98. BenchPress: a man is lifting weights
99. Haircut: a person is getting a hair cut
100. TennisSwing: a woman hits a tennis ball