Beyond Verbs: Understanding Actions in Videos with Text

Shujon Naha
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada
Email: shujon@cs.umanitoba.ca

Yang Wang
Department of Computer Science
University of Manitoba
Winnipeg, MB, Canada
Email: ywang@cs.umanitoba.ca

***Abstract*—We consider the problem of joint modeling of videos and their corresponding textual descriptions (e.g. sentences or phrases). Our approach consists of three components: the video representation, the textual representation, and a joint model that links videos and text. Our video representation uses the state-of-the-art deep 3D ConvNet to capture the semantic information in the video. Our textual representation uses the recent advancement in learning word and sentence vectors from large text corpus. The joint model is learned to score the correct (video, text) pairs higher than the incorrect ones. We demonstrate our approach in several applications: 1) retrieving sentences given a video; 2) retrieving videos given a sentence; 3) zero-shot action recognition in videos.**

I. Introduction

Video understanding is one of the key research problems in computer vision. Most previous research in this area focuses on action classification, where the goal is to assign a video into one of several predefined action categories. In the literature, action classification is often performed by using standard classifiers (e.g. SVM) together with various video-based features, such as spatial temporal interest points [1], motion trajectories [2], deep learning based features [3]. Early work on action recognition typically uses video datasets that are collected in controlled environments (e.g. KTH dataset [4], Weizmann dataset [5]). These videos tend to have simple actions and clean background. Over the year, researchers have introduced more challenging and diverse datasets, e.g. Hollywood-2 [6], UCF actions [7], HMDB [8].

Most previous work in action recognition assumes a fixed set of action labels. However, videos in the real world tend to have much more diversity. Given a video, it is possible to describe the video in several different ways. Figure 1 shows some example videos from the YouTube dataset [9]. Each video is associated with multiple sentence descriptions. We can see that people often use different words to describe the same event in the video. So it is difficult to pre-define a set of discrete category labels that encompass the entire semantic space of videos in the wild.

In this paper, we consider video understanding with textual descriptions. Many online videos (e.g. YouTube, Facebook) naturally come with textual descriptions in various forms. Some of the descriptions can be simple tags, while others can be complex sentences or even paragraphs. These videos provide rich datasets for research in video understanding.

Our training data consist of videos and textual annotations. Each training video can be associated with one or more textual annotations. Previous work in video-based action recognition only considers annotations in the form of single verbs (e.g. "running", "walking", etc). In this paper, we move beyond single verbs and consider textual annotations in richer forms, such as short phrases (e.g. "playing guitar") or full sentences (e.g. "Several teams are playing soccer."). Our goal in this paper is to learn a scoring function to measure the compatibility between a video and a textual description (phrase or sentence) from the training data. During testing, we can use the learned scoring function to measure the compatibility of an unseen video and an unseen textual annotation.

We demonstrate our approach in several applications. One application is sentence ranking/retrieval for videos. During testing, we are given an unseen video as the query. Our goal is to rank textual descriptions (e.g. sentences) according to their relevance to the query video. Compared with standard video classification, our approach has several advantages. First, we do not assume a fixed set of class labels. This allows our approach to potentially handle a very large semantic space. Second, the words use in the textual descriptions do not necessarily have to appear during training. Third, our method returns a ranked list of textual annotations, so we can handle the case that a video can be explained by two different annotations. Similarly, our approach can also be used to rank/retrieve videos given a query sentence. Another application is zero-shot video classification. In this case, we want to classify a video into one of the pre-defined categories. Each category is associated with a short phrase, e.g. "riding horse" "playing violin", etc. But in our case, we assume the category labels in training and testing data are disjoint.

II. Related Work

Our work is related to several areas of computer vision. In this section, we review the work most relevant to us.

Action recognition in videos: Action recognition in videos has been widely studied in computer vision. Most work focuses on classifying videos into a pre-defined set of action

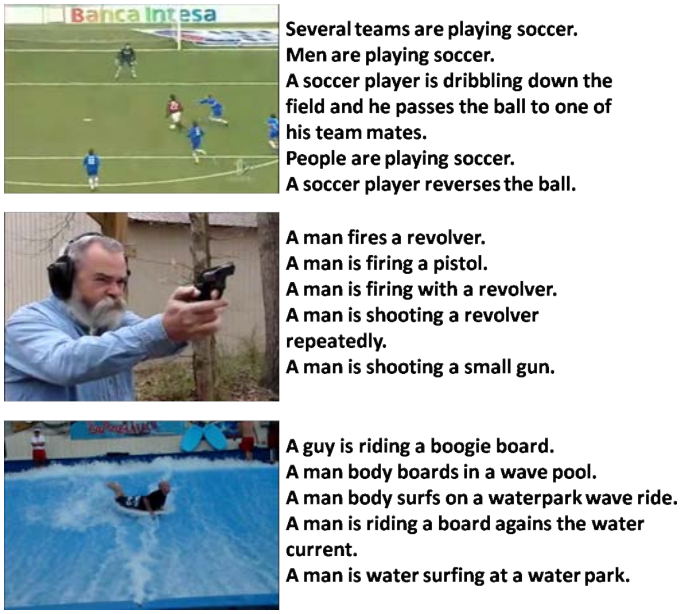

Fig. 1. Examples from the Youtube dataset [9]. Each video is associated with several sentence descriptions. Notice that different verbs can be used to describe the same video (e.g. "play" versus "pass", "fire" versus "shoot", "ride" versus "board" in these examples).

categories, e.g. [5], [10], [1], [2]. Early work in action recognition usually uses hand-designed features, such as STIP [1], dense trajectories [2]. Inspired by recent success of deep learning in object recognition, some recent work [10], [3] uses convolutional neural network to learn the features.

Zero-shot learning: The goal of zero-shot learning is to recognize classes without training examples. In computer vision, attribute-based representation is a popular approach for zero-shot recognition. Farhadi et al. [11] and Lampert et al. [12] use attributes as the intermediate representation shared by object classes. Liu et al. [13] use similar attribute-based representation for action recognition in videos. There is also work on zero-shot event detection [14], [15].

Some recent work uses semantic word embedding (i.e. word vectors) for zero-shot learning in computer vision. Word2vec [16] and Glove [17] are two popular methods for learning word vectors from large corpus. The learned word vectors can be used to measure the semantic distance between two words. If two words (e.g. "dog" and "cat") are semantically close, their word vectors tend to be similar as well. Word vectors have been used for zero-shot recognition of object classes in images [18], [19], [20], [21].

Vision and text: Our work is related to a line of research on connecting videos and text. Guadarrama et al. [9] learn to describe arbitrary activities in videos. Their method exploits the semantic hierarchies of subjects, verbs and objects. Yu et al. [22] learn word meanings with the help of video clips. The closest work to ours is Xu et al. [23] which learns to embed videos and sentences onto a common semantic space.

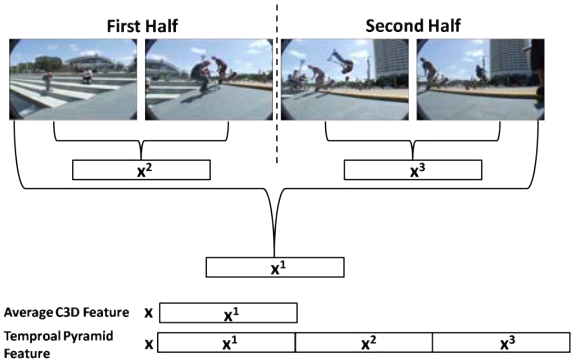

Fig. 2. Illustration of the average C3D feature and the temporal pyramid feature.

III. Our Approach

Let x be a video and y be the textual description (e.g. a sentence). We use $f(x) \in \mathbb{R}^V$ to denote the V-dimensional feature vector extracted from x and $g(y) \in \mathbb{R}^T$ to denote the T-dimensional feature vector extracted from y. Our goal is to learn a matrix $W \in \mathbb{R}^{V \times T}$ of model parameters, so that we can use the following scoring function $S_W(x, y)$ to measure the compatibility of $f(x)$ and $g(y)$:

$$S_W(x, y) = f(x)^\top W g(y) \quad (1)$$

In the following, we describe how to define the video representation $f(x)$, the textual representation $g(y)$, and how to learn the model parameters W.

A. Video Representation

We use the convolutional 3D (C3D) feature proposed in [3] to represent the video. This feature is learned using deep 3-D convolutional network trained on a large scale video dataset. It has been shown to be a state-of-the-art feature representation for video analysis. Given a video, the C3D features are extracted by passing the video through the learned C3D network and taking the output from the $fc7$ layer. C3D extracts a 4096 dimensional feature vector for every 20 frames in the video.

In order to generate a feature vector for the entire video, we have used the following two approaches. Figure 2 illustrates these two features.

Average C3D feature: In this approach, we simply split a whole video x into multiple shots, where each shot consists of 20 frames. The C3D feature is extracted from each shot. We then take the average of the C3D features from all the shots as the feature vector $f(x)$ for the whole video.

Temporal pyramid: The average C3D feature does not take into account the temporal ordering of the shots. For certain events, the temporal ordering might be useful. For example, if a video is described with the sentence description "A young man skateboards along a step, up a ramp, and then does a flip onto the pavement above.", different temporal segments of the video might contain visual information for different actions (i.e. skateboards along a step, flip onto the pavement)

at different temporal locations (Fig. 2). Simply taking the average of the video features from different frame segments is suboptimal since it ignores this temporal structure. To address this issue, we propose another approach that captures some temporal information. We call it *temporal pyramid*. This is inspired by spatial pyramid matching [24]. We first divide the input video into two equal parts. We then compute the average C3D features of the first part, the second part, and the whole video. Finally, we concatenate these three average C3D vectors as the final representation $f(x)$ of the whole video. The dimension of the final feature representation is $4096 \times 3 = 12288$.

B. Textual Representation

Given a textual description y, we define a feature vector $g(y)$ to represent the text. For ease of presentation, we will assume the text description is in the form of a sentence.

We would like the textual feature vector $g(y)$ to have the following property. If two sentences y and y' are similar in terms of their semantic meanings, the distance between the two vectors $g(y)$ and $g(y')$ should be small. If the semantic meanings of y and y' are very different, we would like the distance between $g(y)$ and $g(y')$ to be large.

In this paper, we use two different approaches for the textual representation $g(y)$.

Mean word vectors: Word vectors have recently become a popular representation for capturing semantic meanings. These word vectors are learned from large collections of text documents. In the end, each word is represented as a fixed length vector (also known as *word embedding*). Similar words (e.g. "dog" and "cat") tend to be mapped closer in this embedding space.

Most tools (e.g. word2vec [16] or GloVe [17]) for extracting word vectors only provide the vectors for individual words. But in our case, we need a vector representation for the whole sentence. Our first approach is to simply take the average of the word vectors of all the words in a sentence. In the experiments, we will show that this simple strategy works surprisingly well.

Skip-thought vectors: The skip-thought vector [25] is a very recent work on extracting the vector representation at the sentence level. It was trained from a large collection of more than 11K books. Given a sentence, the skip-thought vector can produce a vector representation that capture the semantic meaning of the sentence. If two sentences are semantically similar, their vector representations will be similar as well.

C. Model Learning

The goal of this section to learn the parameter matrix W. Given the visual representation $f(x)$ and the textual representation $g(y)$, we can score the compatibility of x and y as $S_W(x, y) = f(x)^\top W g(y)$. If the textual description y is a good description of the video x, we would like this score to be high. Otherwise, the score should be low.

We assume a training set with N videos $\{x_1, x_2, ..., x_N\}$. Each training video x_i is associated with one or more textual descriptions (e.g. sentences). We consider these to be the positive descriptions since they are correct textual descriptions of x_i. We use $\mathcal{P}(x_i)$ to denote the set of positive textual descriptions of x_i. We also assume a set of negative textual descriptions for each x_i. A textual description is negative if it is not a good description of the video. We use $\mathcal{N}(x_i)$ to denote the set of negative textual descriptions of x_i. In Sec. IV, we will describe how to construct $\mathcal{P}(x_i)$ and $\mathcal{N}(x_i)$ on the datasets used in the experiments.

Inspired by the large margin criterion in the standard SVM learning, we learn the parameters W by solving the following optimization problem.

$$\min_{W,\xi} \frac{1}{2}||W||_2^2 + \sum_{i=1}^{N} \sum_{j=1}^{|\mathcal{P}(x_i)|} \xi_{ij} \tag{2a}$$

$$\text{s.t.} \quad f(x_i)^\top W g(y_j) \geq f(x_i)^\top W g(y_k) + 1 - \xi_{i,j} \tag{2b}$$

$$\xi_{i,j} \geq 0, \quad \forall i \in \{1, 2, ..., N\}, \tag{2c}$$

$$\forall y_j \in \mathcal{P}(x_i), \forall y_k \in \mathcal{N}(x_i) \tag{2d}$$

For a video x_i, the constraint in Eq. 2b enforces the score of a positive textual description y_j to be higher than that of a negative description y_k by a margin of 1. Similar to standard SVM, the slack variables ξ_{ij}'s are used to allow soft margins. We use stochastic gradient to optimize Eq. 2.

IV. Experiments

We evaluate our approach on two datasets: the YouTube dataset [9] and the UCF-101 dataset [7]. On the YouTube dataset, we consider two applications: retrieving sentences given a video, and retrieving videos given a sentence. On the UCF-101 dataset, we consider zero-shot action recognition.

A. YouTube dataset

The YouTube dataset contains 1970 videos. Each video is associated with multiple sentence descriptions. See Fig. 1 for some examples of this dataset. We only use the sentence descriptions marked as clean in the dataset and ignore the videos that do not have any clean sentence descriptions. We also ignore sentences that do not have a verb. We then split the videos into training and test sets. In the end, the training dataset contains 1300 videos and the test dataset contains 610 videos. On average, each video is associated with 15 sentences.

We learn the model parameters W using the training videos and their associated sentences. For each video, we consider the sentences associated with this video as "positive" textual descriptions. We consider the sentence associated with any other training video as "negative" textual descriptions.

We perform the following two applications on testing videos.

Sentence retrieval: In this task, we are given a query video and try to find good sentence descriptions for this video. For each video in the test set, we rank all the sentences of all testing videos using the scoring function in Eq. 1. Since a video might be associated with multiple correct sentences, we use the Normalized Discounted Cumulative Gains (NDCG) [26] to measure the ranking performance. If all the correct sentences

are ranked higher than the incorrect ones, the NDCG will be high. We report NDCG values at five different truncation levels.

We consider two baselines for comparison. The first baseline (called "verb only") only uses the word vector corresponding to the verb in the sentence as the textual description. In the second baseline (called "bag-of-words"), we create a dictionary of words from the sentences in the YouTube dataset and represent each sentence using the standard bag-of-words representation. In other words, each sentence is represented as a vector of word frequencies in the sentence.

Table I shows the comparison. We consider both the average C3D feature and the temporal pyramid as video representations. We can see that our approach performs much better than the two baselines. This demonstrates the advantage of using a semantic representation of sentences for video understanding. Table I also shows that the temporal pyramid performs slightly better than the average C3D feature.

We also measure the ranking performance using mean rank defined in [23]. For each testing video, we record the rank of the first correct sentence that describes the query video. We then take the mean of the rank over all testing videos to measure the performance. The comparison is shown in Table II.

Figure 3 shows some retrieved sentences for some sample videos.

Video retrieval: In this task, we consider a sentence as the query and retrieve videos described by this query sentence. Our experiment setting is similar to [23]. We randomly select 5 sentences from each of the testing videos. So in total we have 3050 sentences and 610 videos. For each sentence, we rank all the testing videos according to the score defined in Eq. 1. We record the correct video position in the rank for each query. Then we calculate the mean rank over all query sentences to measure the performance of the video retrieval. Table III shows the comparison of different approaches on the YouTube dataset. Again, our approach (skip-though vector, mean word vector) performs much better than the two baselines (verb-only vector, bag-of-words). Similarly, the temporal pyramid feature perform slightly better than the average C3D feature. Figure 4 shows some retrieved videos for some sample sentences.

From the above results, we can see that the skip-thought vector performs the best. If we only use the verb, the performance is very poor. Although verbs can identify the action in the video, it fails to relate the subject and object of the video in the corresponding sentence. For example, given the query sentence "A car is making sharp swerves" (see Fig. 4), the verb-only approach retrieves videos for people and electric spark swerving – the action "swerving" is present, but the subject and the object are totally wrong. Similarly for the query sentence "Two teams are playing soccer", the verb-only approach retrieves video where the "playing" action is present but the subject and the object are completely wrong.

We also compare our results on sentence retrieval and video retrieval with [23]. We have followed the experiment setup in [23] as close as we can, but a direct comparison is difficult since [23] randomly chooses five sentences for each video in the test set. Following [23], we also randomly choose five sentences for each video in the test data. But due to the randomness, the sentences we have chosen are different from those in [23]. With this caveat, Table IV compares our results with [23].

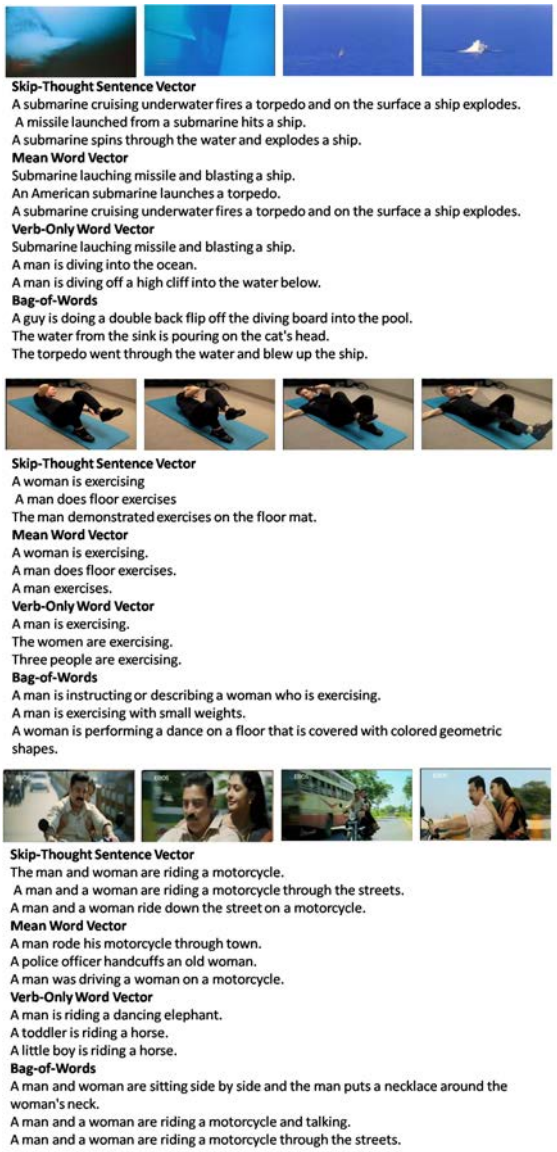

Fig. 3. Top ranked sentences for sample testing videos on the Youtube dataset for different approaches.

B. UCF-101 dataset

The UCF-101 dataset contains 101 action classes and 13320 videos. Each video is associated with a short frame, e.g. "play violin". Most previous work in action recognition uses this dataset for standard action classification.

We perform zero-shot action recognition on this dataset. We split the 101 classes into training and test datasets. The training set consists of 91 classes (we call them "known" classes)

TABLE I

COMPARISON OF MEAN NDCG AT DIFFERENT TRUNCATION LEVELS FOR DIFFERENT APPROACHES TO RANK SENTENCES FOR TEST VIDEOS ON THE YOUTUBE DATASET. OUR APPROACHES (SKIP-THOUGH VECTOR, MEAN WORD VECTOR) PERFORMS MUCH BETTER THAN THE TWO BASELINES (VERB-ONLY VECTOR, BAG-OF-WORDS). IN EACH APPROACH, WE CONSIDER BOTH TEMPORAL PYRAMID AND AVERAGE C3D FEATURES. THE TEMPORAL PYRAMID PERFORMS SLIGHTLY BETTER.

method		NDCG (%) at				
		20%	40%	60%	80%	100%
skip-thought vector	temporal pyramid feature	**31.07**	**32.94**	**33.29**	**33.40**	**33.44**
	average C3D feature	30.63	32.70	33.08	33.26	33.26
mean word vector	temporal pyramid feature	27.66	30.25	30.95	31.13	31.18
	average C3D feature	27.5	29.81	30.49	30.68	30.71
verb-only vector	temporal pyramid feature	21.63	25.25	26.66	27.51	27.74
	average C3D feature	21.12	24.63	26.18	27.09	27.34
bag-of-words	temporal pyramid feature	22.8	26.06	27.48	28.18	28.43
	average C3D feature	21.79	25.26	26.84	27.57	27.83

Fig. 4. Top ranked videos for sample sentences on the YouTube dataset for different approaches.

TABLE II

COMPARISON OF THE MEAN RANK (LOWER IS BETTER) OF DIFFERENT APPROACHES IN THE SENTENCE RETRIEVAL APPLICATION ON THE YOUTUBE DATASET.

method		mean rank
skip-thought vector	temporal pyramid feature	**224.80**
	average C3D feature	231.52
mean word vector	temporal pyramid feature	264.55
	average C3D feature	267.71
verb-only vector	temporal pyramid feature	472.62
	average C3D feature	498.38
bag-of-words	temporal pyramid feature	341.79
	average C3D feature	346.94

TABLE III

COMPARISON OF MEAN RANK (LOWER IS BETTER) OF DIFFERENT APPROACHES IN THE VIDEO RETRIEVAL APPLICATION ON THE YOUTUBE DATASET.

method		mean rank
skip-thought vector	temporal pyramid feature	**59.54**
	average C3D feature	61.63
mean word vector	temporal pyramid feature	81.73
	average C3D feature	83.66
verb-only vector	temporal pyramid feature	112.94
	average C3D feature	113.64
bag-of-words	temporal pyramid feature	117.5
	average C3D feature	120.8

and the test set consists of the remaining 10 classes (we call them "unknown" classes). Since the textual description of a class is a phrase (not a sentence), we cannot apply the skip-thought vectors. So we only apply the mean word vector as the representation of the textual description. Each of the 101 classes is represented as the mean of word vectors in the corresponding phrase. For example, the class "play violin" is represented as the mean of the word vectors of "play" and "violin". For each video, we consider the phrase vector corresponding to its class label as the "positive" textual description. We consider the phrase vectors corresponding to the other 100 class labels as the "negative" textual descriptions. We then learn the model parameters W from the 91 classes in the training set.

For a video in the test set, we use Eq. 1 to calculate the score corresponding to each of the 10 unknown classes. The predicted class label of this video is the one with the maximum

TABLE IV
COMPARISON OF OUR RESULTS WITH [23] IN TERM OF THE MEAN RANK (LOWER IS BETTER) IN VIDEO RETRIEVAL AND TEXT RETRIEVAL APPLICATIONS ON THE YOUTUBE DATASET. *THESE NUMBERS ARE DIRECTLY TAKEN FROM [23]. THE NUMBERS ARE NOT DIRECTLY COMPARABLE DUE TO VARIATIONS IN FEATURES, AND DATASET CONSTRUCTIONS. SEE THE TEXT FOR DETAILS.

Method	Video Retrieval	Text Retrieval
Ours	**149.84**	**83.0**
*Xu et al. [23]	236.27	224.10

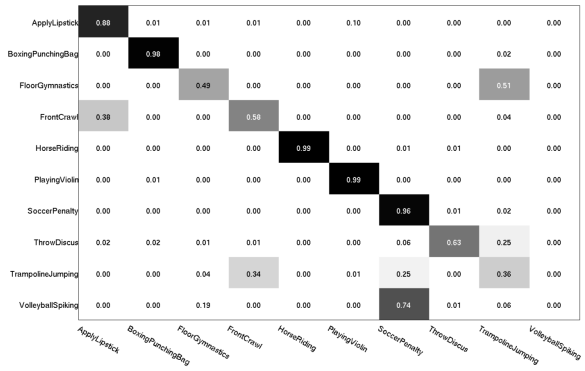

Fig. 5. Confusion matrix of zero-shot recognition on the UCF-101 dataset.

score. Figure 5 shows the confusion matrix of recognizing the 10 unknown classes. We can see that even though we do not have any training data for the 10 unknown classes, we can still correctly recognize most of the classes quite well. The reason is that the model parameters W capture the semantic meaning of words and phrases.

We also compare our zero-shot learning with the IAP method in [12]. The method in [12] uses the attribute-vector associated with each class for zero-shot learning. Since we do not have attributes on the UCF-101 action dataset, we use the mean word vector in [12]. Table V shows the comparison. Our approach outperforms [12] by a large margin.

V. CONCLUSION

Videos provide a rich source of visual data. Most previous work in video understanding focuses on simple action classification. In this paper, we have proposed an approach for learning a joint models of videos and textual descriptions. We have considered textual descriptions in the forms of sentences or phrases. We have demonstrated our approach in several applications: sentence retrieval given a video, video retrieval given a sentence, zero-shot recognition in videos. Our experimental results demonstrate that the proposed method provides an effective way of capturing the relationship of videos and their corresponding linguistic descriptions.

TABLE V
COMPARISON OF OUR APPROACH WITH ZERO-SHOT LEARNING APPROACH IN [12] ON THE UCF-101 DATASET.

method		accuracy(%)
our approach	temporal pyramid feature	**68.50**
	average C3D feature	67.14
Lampert et al. [12]	temporal pyramid feature	27.6
	average C3D feature	27.43

Acknowledgment: This work is supported by NSERC. We thank NVIDIA for the GPU donations used in this work.

REFERENCES

[1] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR*, 2008.
[2] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *ICCV*, 2013.
[3] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015.
[4] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *ICPR*, 2004.
[5] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *ICCV*, 2005.
[6] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *CVPR*, 2009.
[7] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human action classes from videos in the wild," UCF, Tech. Rep. CRCV-TR-12-01, 2012.
[8] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *ICCV*, 2011.
[9] 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 *ICCV*, 2013.
[10] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.
[11] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2009.
[12] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *CVPR*, 2009.
[13] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos"in the wild"," in *CVPR*, 2009.
[14] C. Gan, M. Lin, Y. Yang, Y. Zhuang, and A. G. Hauptmann, "Exploring semantic inter-class relationships (sir) for zero-shot action recognition," in *AAAI*, 2015.
[15] S. Wu, S. Bondugula, F. Luisier, X. Zhuang, and P. Natarajan, "Zero-shot event detection using multi-modal fusion of weakly supervised concepts," in *CVPR*, 2014.
[16] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013.
[17] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vector for word representation," in *EMNLP*, 2014.
[18] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *CVPR*, 2015.
[19] S. Naha and Y. Wang, "Zero-shot object recognition using semantic label vectors," in *CRV*, 2015.
[20] 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," in *ICLR*, 2014.
[21] R. Socher, D. Chen, C. D. Manning, and A. Y. Ng, "Reasoning with neural tensor networks for knowledge base completion," in *NIPS*, 2013.
[22] H. Yu and J. M. Siskind, "Grounded language learning from video described with sentences," in *ACL*, 2013.
[23] R. Xu, C. Xiong, W. Chen, and J. J. Corso, "Jointly modeling deep video and compositional text to bridge vision and language in a unified framework," in *AAAI*, 2015.
[24] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognition natural scene categories," in *CVPR*, 2006.
[25] R. Kiros, Y. Zhu, R. Salakhutdinov, R. Zemel, A. Torralba, R. Urtasun, and S. Fidler, "Skip-through vectors," in *NIPS*, 2015.
[26] O. Chapelle, Q. Le, and A. Smola, "Large margin optimization of ranking measures," in *NIPS Workshop on Learning to Rank*, 2007.