Action2Vec: A Crossmodal Embedding Approach to Zero Shot Action Learning

Meera Hahn
Georgia Institute of Technology
meerahahn@gatech.edu

Andrew Silva
Georgia Institute of Technology
andrew.silva@cc.gatech.edu

James M. Rehg
Georgia Institute of Technology
rehg@gatech.edu

Abstract

This abstract creates a cross-modal embedding space of actions in videos and verbs. Unlike objects, categories of actions are not clear cut. Actions are executed differently by different people and the beginning and end of an action is often left to interpretation of the viewer. Rather than strictly categorize actions with labels it is easier to view them in the context of other actions and to describe them using natural language. In this work we demonstrate a novel method which uses a hierarchical recurrent network to project actions from videos in a visual-semantic space. To quantitatively evaluate our embedding architecture we do zero-shot action learning and achieve state-of-the-art results.

1. Introduction

Many core problems in AI revolve around the successful fusion of language and vision. Language is both the primary modality for human communication and the primary source of information (semantics) about the visual world. In the past decade, progress in object detection and categorization has driven a wide range of work that connects words and images in tasks ranging from image captioning [18] to the construction of joint embedding spaces for images and text [6]. In contrast to these successes in connecting text and images, less progress has made in connecting text and video. There are several potential reasons for this. First, in video the primary lexical unit of meaning is the verb, which defines existence and change. Verbs don't map to regions of pixels in video in the same straightforward way that concrete nouns map to bounding boxes in images. A verb defines a sentence in the same way that an action defines a video– by creating a structure that constrains all other elements. Second, while actions often map to visual movement, in real-world videos there are many sources of movement (camera motion, background objects, etc.) which are not part of the action, and actions can be defined by a relative lack of movement (e.g. sitting and reading). Bringing verbs and video into correspondence therefore requires the ability to integrate information over multiple temporal scales of meaning. This is particularly challenging in light of the relatively poor performance of modern action recognition methods (in comparison to object detection) due to the computational demand and the dimensionality of the data.

In this abstract, we propose a hierarchical LSTM model to generate a joint embedding space for action videos and verbs, thus providing a novel semantic video representation. We establish the utility of action representation for the visual domain by demonstrating that it achieves state-of-the-art results on zero shot action recognition.

2. Related Work

The emergence of accurate distributional semantic models for language has opened the door for advanced multimodal fusion of vision and language. Work on creating joint image-word and image-sentence spaces has been successful at projecting images into the language space and vise versa [1, 2, 6, 12]. Most of the evalutions for these works tested the embeddings by using them for the tasks of image captioning, retrieval, and generation.

While the above joint language and vision representations have focused on images, there has been work on bridging the gap between videos and language for the task of video captioning [10, 11]. These efforts have shown promising results, however the focus of the research has been on decoding accurate captions rather than producing a robust encodings of videos. In these approaches, the target videos are encoded by performing feature extraction using networks such as C3D [14], and then performing temporal pooling of all frame feature vectors to obtain a single vector to represent the entire video. While this encoding scheme was effective for producing accurate captions, there is reason to believe that a more fine-grained video representation based on recurrent network models could provide a superior representation.

A few works have used word embedding for the purpose

of zero-shot action recognition [7, 15, 16, 17]. Zero-shot action recognition is where actions or activities that have not been previously seen by a model can be classified based on their mapping into a semantic space. Language is the most obvious and easy semantic space to map visual actions into because it is the one that we, as humans, use to ground our visual perception to meaning. Additionally, we can take advantage of extremely informative word embeddings from recent advances in distributional semantics. All of the prior works in this area do not utilize video embedding representations based on deep feature learning, but instead use low level video features such as Histogram of Gradients and Motion Boundary Histogram to encode the video [15]. These works all use ridge regression to map the video encoding into the Word2Vec feature space, and focus on alleviating the domain shift problem that occurs in zero-shot action learning [16]. The domain shift occurs when switching from seen to unseen data. A regression mapping is fitted to the data distributions for training classes, which causes lower performance on test classes. There are many efforts to ameliorate this problem, however most include using auxiliary datasets or self training, as in [15], which requires having access to the knowledge of which classes are in the test set. While this is an effective solution, it runs contrary to the defining principle of zero shot learning. In contrast, we define our zero-shot learning problem as restricting access of labeled videos to only the training set and having full access to all words and phrases in the semantic set. We approach the domain shift problem through regularization using the unlabeled testing set.

In order to obtain high-quality action embeddings, we train our model using supervision from action recognition and word embeddings. We build on many recent action recognition approaches. In particular, we make heavy use of C3D [14] features throughout our work.

3. Approach

The goal of this work is the creation of semantically rich embeddings of actions. In other words, we want encode each video into a dense embedding. To do this, we combine the spatio-temporal visual information of the video with the linguistic meaning of the action's corresponding verb. To obtain the linguistic information for each action class name, we use the word embeddings created by Mikolov et al.'s Word2Vec skip-gram model with negative sampling [9]. Word2Vec creates distributional representations for words which have been shown to be highly accurate when used as features representations in NLP tasks. The Word2Vec model we use in this work is trained on the Google News Dataset[1]. This model contains 300 dimensional vectors for approximately 3 million English words.

The first step of extracting the visual information from our videos is obtaining frame level features. Following the success of C3D features for action recognition [11, 14], we extract C3D feature vectors for each 16th frame of the videos. To handle the varying length of videos, we set a max number of feature-vectors of 21 and pad sequences that are under 21 in length with zeros. After the feature generation phase, every video is represented by 21 feature vectors, each with 4096 dimensions. The C3D model we use is pre-trained on Sports-1M dataset [3].

Once we have our videos encoded as a sequence of 21 C3D vectors, we pass them through a 2-layer LSTM model, visualized in Figure 1. We were motivated to use hierarchical recurrent networks, as they have been shown to perform well in video captioning [10]. First, we divide our the 21 C3D vectors into 7 non-overlapping sub-sequences of length 3. Each sub-sequence is then passed through an LSTM with 1024 hidden units and dropout of 0.5, which outputs a 1024-dimensional vector for each input sequence. This LSTM is shared between all 7 inputs, so the weights do not change depending on which input is being processed, and the hidden states do not carryover between forward passes. Each input is treated as independent in the sequence, and the first LSTM transforms each of the 3x4096 input into a 1x1024 vector

After the first LSTM, all 7 outputs are concatenated into a single 7x1024 vector. This is passed through a second LSTM, with 512 units and no dropout, which outputs a 512-dimensional vector. The output of the second LSTM passes through a 300 dimensional fully-connected layer, which is the same dimensionality as our word embedding labels. This vector is directly compared against the word embeddings using the pairwise ranking loss from [6]. For our cross-entropy classification loss, the 300-dimensional vector goes through one final fully-connected layer with the same dimensionality of our one-hot labels, and with a sigmoid activation. The entire network is trained with a categorical cross-entropy loss between the predicted and actual classes, and the pairwise-ranking loss between our 300-dimensional embedding and the word vectors for the labels. The network is optimized with Adam [5]. For simplicity, from here on we refer to our architecture as Action2Vec.

Action2Vec uses dual loss shown in Eq. 2, which combines a pairwise-ranking loss, $\mathcal{L}_{CE}$, and a cross-entropy classification loss, $\mathcal{L}_{CE}$.

$$L_{PR} = \min_{\theta} \sum_{i} \sum_{x} (1 - s(a_i, v_i)) + max\{0, s(a_x, v_i)\} + max\{0, s(a_i, v_x)\} \quad (1)$$

$$L_{dual} = L_{PR} + \alpha * \mathcal{L}_{CE} \quad (2)$$

Where v_i is a verb embedding of class i, a_x is an action-video embedding of contrastive class k, and v_x is a con-

[1] https://code.google.com/archive/p/word2vec/

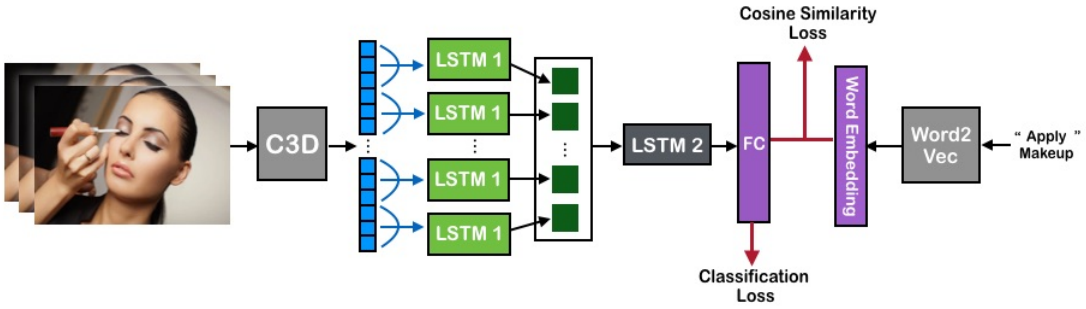

Figure 1. Hierarchical LSTM architecture for embedding videos into a 300-dimensional vector for zero shot learning. Videos are passed through C3D which outputs a 4096-dimensional vector for each 16-frame segment of a video, and our 'LSTM 1' layer outputs a 1024-dimensional vector for each sequence of C3D feature vectors. These vectors are concatenated and passed through the 'LSTM 2' layer and a fully-connected layer of size 300 before going to the two loss functions.

trastive verb embedding of class k. We use cosine similarity as our similarity function s.

We use the word vectors of class names as labels for the pair-wise ranking loss. Certain class names are not in verb form, so we edit the names to an equivalent word that exists in the Word2Vec model. For example the class name "walking" is a adjective and noun, so it is changed to the verb "walk". Similarly, the class name "clean and jerk" was not in the Word2Vec model, so the name was changed to the analogous name of "weightlift." For class names that are longer than a single word, we average the word vectors that make up the name.

4. Evaluation

In this section we validate the quality of our networks ability to project new videos into the joint space by running zero shot action recognition and obtaining state of the art results. Our zero-shot learning setup takes a labeled training set and unlabeled testing set. The labels of the training set and testing set have no overlap, and the labels of the testing set are never seen by any model in our experiment. First, we train our hierarchical recurrent network (HRN) described in §3 on the videos and labels of our training set. We then use the trained HRN to encode all videos in the test dataset. The fully connected layer of the HRN is extracted as the latent vector representation for the video. These predicted video vectors are then normalized. Then each predicted vector in test set is then assigned the label of the nearest verb embedding. Nearest neighbors are calculated using cosine distance. The accuracy of the zero shot method is calculated based on the number of test videos that had their action class predicted correctly.

We test on the common action recognition datasets: HMDB51 [8], UCF101 [13] and Kinetics [4]. For each dataset, we test on three different amounts of held out data: 50%, 20% and 10%. We observe that performance decreases as we withhold a greater number of classes, which is expected because the model has less information with which to understand how to interpret new action classes. To test these splits fairly, we randomly generate 3 trials of each split type. Table 1 shows the average accuracy of the trials.

height	**HMDB51**	**UCF101**	**Kinetics**
50/50			
Action2Vec	**24.10**	**21.96**	**15.34**
Pooled C3D fc7	5.0	11.41	9.89
Xu. et al [15]	15.0	15.8	-
Kodirov. et al [7]	-	14.0	-
80/20			
Action2Vec	38.16	37.39	25.59
Pooled C3D fc7	8.77	23.89	18.76
Kodirov. et al [7]	-	22.5	-
90/10			
Action2Vec	51.85	49.44	36.88
Pooled C3D fc7	23.11	36.29	26.31

Table 1. Accuracy for zero-shot action recognition on the HMDB51, UCF101 and Kinetics datasets. The left most hand column refers to the methodology used. The numbers here are the percentage of test videos that were classified with the correct class name. The datasets are broken into three different splits of training and test data. For each split, the class names of the train and test set are completely disjoint.

4.0.1 Zero-Shot Baselines

The first baseline is a zero-shot recognition method from Xu. et al [15], which maps actions into the word semantic space. This method uses low-level features such as HOG to create the video embeddings and then trains a Kernel Ridge Regression model to map the action space to the semantic space. Like Action2Vec, they use the Word2Vec word embeddings as their action semantic space. To deal with the regression domain shift problem, this baseline uses "self-training," which readjusts the word embeddings for the testing classes during testing. This adjustment requires access to the class names of the test set, which conflicts with the

definition of zero-shot defined in this paper. However, it is still a good baseline because it uses a mapping from actions to words to perform the zero-shot recognition. The second baseline is Kodirov. et al [7] which also uses low-level features for the video representation, but takes a unsupervised approach to dealing with domain shift. The final baseline we compare to is pooled C3D features as video embeddings. We take the C3D features that we extracted for every 16th video frame and average them. This is a common representation used in video captioning papers [11]. We train a Kernel Ridge Regression model with Laplacian regularization to map the pooled C3D vectors to the word vector labels. All results are given in Table 1.

4.0.2 Zero-Shot Analysis

In Table 1 we can observe that Action2Vec embeddings outperform all baseline methods in every data split, for every dataset. We most likely surpass the performance of the two previous zero-shot methods since our model uses deep video features and a neural network to train the regression mapping to the semantic space. We perform slightly better on the HMDB51 dataset than we do on UCF101, which is interesting because HMDB51 is smaller dataset with greater scene diversity. Superior performance on a smaller and harder dataset is a good indicator that Action2Vec is capturing meaningful semantic and temporal data. We can conclude this because the other methods, in particular the pooled C3D baseline, perform significantly worse on HMDB51 than on UCF101. While pooled C3D features provide useful visual descriptors for an embedding, the approach still lacks the longer temporal modeling capacity of a hierarchical recurrent network. This demonstrates the importance of using a recurrent method to capture and encode the temporal information that may be lost when strictly using 3D convolutions.

5. Conclusion

The fusion of perception systems is increasingly integral to the move toward more advanced intelligent systems. Within this, the drive to work on real world, temporal data requires the video domain to be properly explored and accurately represented. In this abstract we posit that the proper way to represent videos is to describe them in the context of verbs. We introduce an architecture which combines the spatial and temporal information from video frames and the semantic information from verbs. We then go on to demonstrate the power of this architecture by using it to recognize previously-unseen actions in the task of zero-shot learning. With these high level video representations, we can work toward solutions for unsolved tasks such as video generation and video question answering.

References

[1] K. Chen, C. B. Choy, M. Savva, A. X. Chang, T. Funkhouser, and S. Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. *arXiv preprint arXiv:1803.08495*, 2018. 1

[2] J. Dong, X. Li, and C. G. Snoek. Word2visualvec: Image and video to sentence matching by visual feature prediction. *CoRR*, 2016. 1

[3] 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. 2

[4] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 3

[5] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[6] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014. 1, 2

[7] E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Unsupervised domain adaptation for zero-shot learning. In *ICCV*, 2015. 1, 3, 4

[8] H. Kuehne, H. Jhuang, R. Stiefelhagen, and T. Serre. Hmdb51: A large video database for human motion recognition. In *High Performance Computing in Science and Engineering 12*, pages 571–582. Springer, 2013. 3

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 2

[10] P. Pan, Z. Xu, Y. Yang, F. Wu, and Y. Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *CVPR*, 2016. 1, 2

[11] Y. Pan, T. Mei, T. Yao, H. Li, and Y. Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016. 1, 2, 4

[12] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, and A. Torralba. Learning cross-modal embeddings for cooking recipes and food images. *CVPR*, 2017. 1

[13] 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. 3

[14] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 1, 2

[15] X. Xu, T. Hospedales, and S. Gong. Semantic embedding space for zero-shot action recognition. In *ICIP*, 2015. 1, 2, 3

[16] X. Xu, T. Hospedales, and S. Gong. Transductive zero-shot action recognition by word-vector embedding. *International Journal of Computer Vision*, pages 1–25, 2017. 1, 2

[17] X. Xu, T. M. Hospedales, and S. Gong. Multi-task zero-shot action recognition with prioritised data augmentation. In *ECCV*, pages 343–359. Springer, 2016. 1

[18] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo. Image captioning with semantic attention. In *CVPR*, pages 4651–4659, 2016. 1