Action2Vec: A Crossmodal Embedding Approach to Action Learning

Meera Hahn
meerahahn@gatech.edu
Andrew Silva
andrew.silva@gatech.edu
James M. Rehg
rehg@gatech.edu

Georgia Institute of Technology
Atlanta, USA

Abstract

We describe a novel cross-modal embedding space for actions, named *Action2Vec*, which combines linguistic cues from class labels with spatio-temporal features derived from video clips. Our approach uses a hierarchical recurrent network to capture the temporal structure of video features. We train our embedding using a joint loss that combines classification accuracy with similarity to Word2Vec semantics. We evaluate Action2Vec by performing zero shot action recognition and obtain state of the art results on three standard datasets. In addition, we present two novel analogy tests which quantify the extent to which our joint embedding captures distributional semantics. This is the first joint action embedding space to combine text and action videos, and the first to be thoroughly evaluated with respect to its distributional semantics.

1 Introduction

Many core problems in AI hinge on the successful fusion of language and vision. Language is both the primary modality for human communication and the primary source of meaning (semantics) for the visual world. In the past 15 years, progress in object detection and categorization has driven a wide range of work that connects words and images in tasks ranging from image captioning [2, 31, 37] to the construction of joint embedding spaces for images and text [5, 16, 25]. In contrast to the success 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 a verb, which defines existence and change. Verbs don't map to regions of pixels in video in the same straightforward way that 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 text and video into correspondence therefore requires the ability to integrate information over multiple spatial and temporal scales of meaning. Such an integration is particularly important in modeling *actions*, as text and video provide complementary sources of information: text encodes the semantics while video provides information about how the action is performed, through the movement patterns it encodes.

Modern deep architectures for action recognition achieve spatio-temporal integration of information by learning spatio-temporal features from a dataset so as to maximize the prediction accuracy for action labels. While this approach has yielded significant gains in recognition performance on real-world datasets, it has two disadvantages. First, the resulting model is highly-specialized to the chosen dataset. Second, the learned representation cannot be easily repurposed for tasks other than the targeted prediction problem. In contrast, prior work on learning image and text *embeddings* has yielded representations which are both more flexible (e.g. can be applied to novel prediction problems via zero shot learning) and capable of providing insight into the structure of the representation (e.g. via the support for vector space semantics [5, 16, 25]). This paper describes a joint embedding space for actions, which we call *Action2Vec*, which is generated by a hierarchical LSTM model, and which links videos and verbs and provides a novel *semantic action representation*. Specifically, Action2Vec takes a video clip and associated text label (typically a single verb or a verb+noun pair) and maps it to a vector that jointly encodes both the semantic and visual properties of the action.

We show that Action2Vec provides a useful discriminative representation for actions by demonstrating its ability to achieve state-of-the-art results for zero-shot learning on several major action datasets. We also show that the Action2Vec embedding preserves locality, in the sense that actions that are visually and semantically similar lie close together in the embedding space. We conducted two novel analogy experiments to evaluate the structure of the embedding space and assess their accuracy as a generative representation of actions. First, we use the standard linguistic lexicon WordNet to test the distribution of vectors in our cross-modal (text + video) embedding space and compare it to the Word2Vec embedding space. By comparing different embedding techniques in the form of confusion matrices to WordNet, we are able to test the accuracy and quality of the embeddings for all verbs, something which has not been done before. Second, we evaluate the distributional properties of the embedding space using vector space arithmetic. For example, given two action classes that share the same verb but utilize different nouns, such as "play piano" and "play violin," we perform the operation: action2vec(play piano) - word2vec(piano) + word2vec(violin) to yield a novel action descriptor. We show that this descriptor vector is closest to the cross modal embedding for "play violin." Vector arithmetic demonstrates that the multi-modal distributed embedding representation that we have produced retains the semantic regularities of word embeddings. Our results demonstrate that Action2Vec provides a flexible and interpretable representation of actions in video.

This paper makes four contributions. First, we present a method for generating a joint visual semantic embedding of video and text which we refer to as Action2Vec. As part of this work, we will release both our software for learning embeddings and the trained embeddings we generated for all of the major action recognition datasets. Second, we demonstrate the quality of the resulting embeddings by obtaining state-of-the-art results for zero shot recognition of actions, and in addition introduce an experiment design which avoids some problems in addressing domain shift that arose in prior works on zero shot action recognition. Third, we introduce a new way to test the semantic quality of verb embeddings through the use of confusion matrices. Fourth, we use vector arithmetic to verify the distributional properties of our embeddings and obtain the first vector space results for actions.

2 Related Work

There has been extensive work in computer vision and machine learning that addresses the task of learning embeddings and the closely-related problem of manifold learning. This lit-

erature is too extensive to review here, and so we focus our attention on works that construct embeddings of text and images. One body of related work, which we utilize in our solution approach, is the construction of distributional semantic models of language in the form of word embeddings [19, 20, 24]. Mikolov et al. [19] is a representative and widely-used example. They introduce a skip-gram-based model to map words into a low-dimensional dense descriptor vector. They demonstrate its ability to perform analogical reasoning and support compositionality. Our work can be seen as an extension of this approach which leverages video features to construct a joint embedding. Our learning architecture is substantially different from [19], due to the unique aspects of learning from video.

The emergence of accurate distributional semantic models for language led in turn to works that created joint image-word and image-sentence embedding spaces [5, 6, 10, 16, 25, 31, 32]. These models can be used to project images into the language space and vice versa. Of these efforts, the paper by Kiros et. al. [16] is perhaps the closest to our work, in that they demonstrate a joint visual-text embedding space that supports vector arithmetic. Specifically, they demonstrate that the image vector for a "blue car" minus the word vector for "blue" plus the word vector for "red" results in a vector that lies in the space of images of red cars. These findings inspired our own vector arithmetic experiments. Our architecture and training approach differ significantly from [16], due to the fact that we are constructing representations from videos instead of still images. A goal of our work is to explore trade-offs between representations that are purely discriminative, optimizing for classification accuracy, and representations that capture the semantic structure of the verb space. We achieve this by optimizing a dual loss that combines a classification loss with a cosine loss that enforces similarity to a Word2Vec embedding (see Figure 2). Other works which have utilized such a dual loss include [5, 25].

Efforts to bridge the gap between *videos* and language have primarily in the context of video captioning [22, 23]. While these efforts have shown promising results, their focus is on generating accurate captions rather than producing a semantically-structured video encoding. In these works, the target videos are encoded by performing feature extraction using networks such as C3D [30] and VGG16 [28], and then performing temporal pooling of all frame feature vectors to obtain a single vector that represents the entire video. While this encoding scheme is capable of producing accurate captions, our approach is motivated by the belief that a more fine-grained video representation based on recurrent network models can provide a superior representation. Our experimental findings demonstrate the benefits of a more fine-grained recurrent video encoding.

Zero-shot learning is a canonical task for evaluating the effectiveness of an embedding space, and several works have used word embeddings to tackle the problem of zero-shot action recognition [17, 34, 35, 36]. The task of zero-shot recognition is to predict the category label for a novel action, which was not known at training time, by mapping it via a previously-trained semantic space. Prior works have used the Word2Vec feature space in conjunction with low-level predefined feature representations such as Histogram of Gradients and Motion Boundary Histogram to encode the video [34]. Specifically, most of these works use ridge regression to map the video encoding into the Word2Vec feature space, and focus on alleviating the resulting *domain shift* problem [36]. The domain shift occurs when switching from seen to unseen data. In the context of zero shot learning, the unseen test classes are often poorly-explained by the regression mapping that is learned from the training distribution, leading to poor test performance. There are many efforts to ameliorate this problem, however most require the use of auxiliary datasets or self-training. An example is [34], which requires access to the knowledge of which classes are in the testing set. While

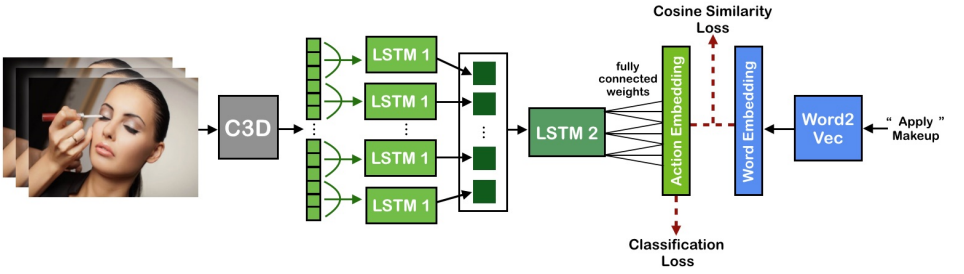

Figure 1: Hierarchical LSTM architecture for embedding videos into a 300-dimensional vector. Videos are passed through C3D which outputs a 4096-dimensional vector for each 16-frame segment of a video, and our 'LSTM 1' layer outputs a 1024-dimensional vector for each sequence of C3D feature vectors. These 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.

this is an effective solution, it runs contrary to our definition of zero-shot learning by having access to the labels of the test set. In contrast, we approach the domain shift problem through regularization using the unlabeled testing set.

The final body of related work is recent deep learning approaches that construct video representations for supervised prediction tasks such as action recognition. We build on these approaches in our own work. In particular, our model utilizes C3D [30] features heavily. We also experimented with two-stream approaches similar to [4, 7, 27], although in our application we found only minimal benefit from the additional network structure.

3 Approach

Our goal is to develop a vector descriptor for an action that captures both the semantic structure of the space of verbs and the spatio-temporal properties that characterize an action in video. We believe we are the first to construct an action embedding that marries features from language and video, and we argue that this construction is the key to developing a comprehensive action model which supports both reasoning and classification tasks. Reasoning refers to the ability to make analogies and perform compositions, and otherwise explore the "what" of an action (e.g. what it is for, what are the situations in which it is used, etc.) In contrast, the spatio-temporal pattern of an action captures the "how" of an action (e.g. how is the action performed, how does it transform the world, etc.) and supports classification. Our approach to constructing the Action2Vec encoding is to obtain linguistic information from action verb names, and combine it with visual information derived from deep video features. To obtain the linguistic information for each action class name, we use the word embeddings created by the Word2Vec skip-gram model with negative sampling [19]. 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.

To extract the corresponding visual information about each action from videos we use frame level C3D features, given their success in action recognition [23, 30]. We extract C3D

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

feature vectors for each 16th video frame. 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. We use a C3D model which was pre-trained on the Sports-1M dataset [11]. Once we have our videos encoded as a sequence of 21 C3D vectors, we pass them through a 2-layer LSTM model, illustrated in Figure 2. We were motivated to use hierarchical recurrent networks, as they have been shown to perform well in video captioning [22]. First, we divide 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 [16]. 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 a pairwise-ranking loss between our 300-dimensional embedding and the word vectors for the labels. The network is optimized with Adam [14]. For simplicity, from here on we refer to our architecture as Action2Vec. Action2Vec uses a dual loss, shown in Eq. 2, that combines a pairwise-ranking loss, $\mathcal{L}_{PR}$ and a cross-entropy classification loss, $\mathcal{L}_{CE}$. The pairwise-ranking loss is shown in Eq. 1.

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

$$\mathcal{L}_{Dual} = \mathcal{L}_{PR} + \alpha * \mathcal{L}_{CE} \tag{2}$$

Where a_i is an action-video embedding of class i, 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 contrastive verb embedding of class k. We use cosine similarity as our similarity function s, and set α to 0.02 after 75% of the iterations of the first epoch.

We use the word vectors of class names as labels for the pairwise-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 present the results from our extensive evaluation of the Action2Vec embedding, consisting of two main parts. First, in §4.2 we validate the quality of our network's ability to project new videos into the joint space by performing zero shot action recognition

and obtaining state of the art results. Second, in §4.3 we present *two novel forms of analogy tests* to validate the semantics of our representation. In the first test, we construct confusion matrices using the similarities between verbs based on Action2Vec and Word2Vec, and then compare the confusion matrices using ranking correlation. This novel approach allows us to directly rate the distances for all verbs in the vocabulary, something that has not been done previously. Our second test performs a systematic and thorough evaluation of vector arithmetic for action embeddings, a topic which has not been explored in previous works. Even within the NLP literature, the standard testing sets for verbs are limited to verb tenses [1, 8]. Collectively, these experiments constitute the most thorough evaluation of verb embeddings that has ever been performed.

4.1 Datasets

Our evaluations are based on three standard datasets for action recognition: UCF101 [29], HMDB51 [18] and Kinetics [12]. We selected these datasets because they contain diverse videos for each action class, allowing us to test the generalization of our method to actions in various environments, poses, and contexts. For example, in the HMDB51 dataset, the action class "push" has a variety of videos, from children pushing toy trains to adults pushing tables. All datasets are focused on human activities from human-human interaction, human-object interaction, pure body motion, and playing musical instruments and sports. UCF101 has a total of 13,320 clips in the dataset with an average clip length of 7.21 seconds. HMDB51 has 51 action classes with a total of 6,776 clips. The recently-released Kinetics dataset is one of the largest action datasets with 400 action classes and a total of 306,245 clips with an average length of 10 seconds. This paper is the first to use Kinetics for zero-shot learning. We use Kinetics to show that our embeddings can scale to larger datasets as well as to obtain the most diverse set of embeddings by using a dataset with a large number classes.

4.2 Zero-Shot Action Recognition

Our learning setup for zero-shot action recognition begins with 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 Action2Vec as described in §3, using cosine distance loss, on the videos and verb embeddings of our training set. We then use the trained model to encode all videos in the test dataset. These predicted video vectors are then normalized and 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 datasets described above: HMDB51, UCF101 and Kinetics. 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.

4.2.1 Zero-Shot Baselines

The first baseline is a zero-shot recognition method from Xu. et al [34], 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.

	HMDB51	UCF101	Kinetics
50/50			
Action2Vec	**22.39**	**21.63**	**15.35**
Pooled C3D fc7	5.00	11.41	9.89
Xu. et al [34]	15.0	15.80	-
Kodirov. et al [17]	-	14.00	-
80/20			
Action2Vec	**39.85**	**35.82**	**22.34**
Pooled C3D fc7	8.77	23.89	18.76
Kodirov. et al [17]	-	22.50	-
90/10			
Action2Vec	**58.01**	**47.94**	**36.93**
Pooled C3D fc7	23.11	36.29	26.31

Table 1: Classification accuracy for zero-shot action recognition on standard datasets.

This adjustment requires access to the class names of the test set, which conflicts with the definition of zero-shot recognition used in this paper. However, it is still a useful baseline because it uses a mapping from actions to words to perform the zero-shot recognition. The second baseline is by Kodirov. et al [17]. It also uses low-level features for the video representation, but takes a unsupervised approach to addressing domain shift. Our final baseline uses pooled C3D features to construct the 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, such as [23]. 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.

4.2.2 Zero-Shot Analysis

In Table 1 demonstrates that the Action2Vec embeddings outperform all baseline methods in every data split, for every dataset. Our use of deep features and neural network regression is one likely reason. Our better performance on HMDB51 compared to UCF101 is interesting as HMDB51 is smaller and has greater scene diversity. Superior performance on a smaller and harder dataset suggests that Action2Vec is capturing meaningful semantic and temporal properties. Additional support comes from the poor performance of the pooled C3D baseline on HMDB51 vs. UCF101. The pooled C3D features provide useful visual descriptors, but lack the extended temporal modeling capacity of the hierarchical recurrent network. This demonstrates the importance of using a recurrent model.

4.3 Analogy Tests

In natural language processing, representations of distributional semantics are commonly evaluated using analogy tests, which take the form: $vector_1 - vector_2 + vector_3 = vector_4$. The test is passed if the word corresponding to $vector_4$ makes logical sense. For example, in the analogy $King - Man + Woman$, the resulting vector should represent the word *Queen*. Analogy tests are the gold standard for evaluating distributional semantics because they assess the relational capacity of the vector space. There are a few standard manually constructed analogy test sets [1, 8, 19]. Unfortunately, these test sets are not comprehensive, covering nouns thoroughly but not adequately testing verbs. In fact, there are no existing test sets whose verb coverage is sufficient, as current evaluations only test the ability to translate between verb tenses. An example of an standard test is "accept is to acceptable as achieve

	HMDB51	UCF101	Kinetics
WordNet vs. Pooled C3D	0.0716	0.0773	0.0929
WordNet vs. Word2Vec	0.2855	0.2292	0.2807
WordNet vs. Action2Vec	0.2353	0.2092	0.2421

Table 2: Spearman Rank Correlation between the gold standard WordNet similarity confusion matrix and the confusion matrices created based on embeddings.

is to what," with the answer being "achievable" [8]. To address this issue, we *introduce two new methodologies* for conducting verb analogy tests which can scale to any dataset.

4.3.1 Word Matrices

WordNet is a large and popular English lexical database that contains the majority of words in the English language[21]. The words in this database are grouped into synsets, which capture both lexical and semantic relations and form a graph of words. Using the synsets, we can measure the relations between individual words. Specifically, the Wu-Palmer algorithm can be used to measure the semantic similarity between any two words in WordNet [33]. Relational distances between words in WordNet are a kind of ground truth distance, as WordNet was manually constructed by linguists.

We now describe our novel construction of word similarity matrices and their use in evaluating our embedding: For each dataset, we take the class name list and remove all names that are not in WordNet. Then for each class name list we create confusion matrices, where the values of the matrix at a given index corresponded to the Wu-Palmer WordNet similarity distance [33]. Then we take the corresponding Action2Vec embeddings for the class name list and create the same confusion matrix, except that now the values of the matrices at a given index corresponded to the cosine similarity between the two Action2Vec embeddings. In order to get a single embedding for an action class, we average the Action2Vec vectors for all videos of that action class. Finally, we create two additional confusion matrices, one using the corresponding Word2Vec embeddings and one using the pooled C3D embeddings.

Taking WordNet to be the ground truth similarity confusion matrix, we compare it to each of the Action2Vec, Word2Vec, and C3D similarity confusion matrices. Since the measurements of cosine similarity and Wu-Palmer similarity are at different scales, we compare the matrices using Spearman Ranking Correlation [13]. We can only calculate ranking correlation between pairs of rows, so we average the ranking correlations across all rows and use that as the ranking correlation between two matrices. Ranking correlation lies between -1 and 1, with 1 being the best possible score and identical ranks. The results of the ranking correlations between all matrices are shown in Table 2.

From Table 2 reveals that the Word2Vec matrix has the highest correlation with WordNet in every dataset. This is to be expected, as Word2Vec has been trained specifically for the task of linguistic representation. In contrast, the pooled C3D vectors are significantly worse than the other two embeddings. This demonstrates how little relational semantic information a purely visual encoding of a video contains. Action2Vec matrix comes in second but not far behind the Word2Vec matrix. This shows us that the Action2Vec architecture, in addition to capturing visual information, is apt at semantically encoding the similarities between actions.

	UCF101	Kinetics
Number of Comparisons	90	1540
Average Precision	0.9875	0.5755

Table 3: Vector arithmetic analogy test results. First row: total number of analogy tests for each dataset. Second row: percentage of tests that were passed.

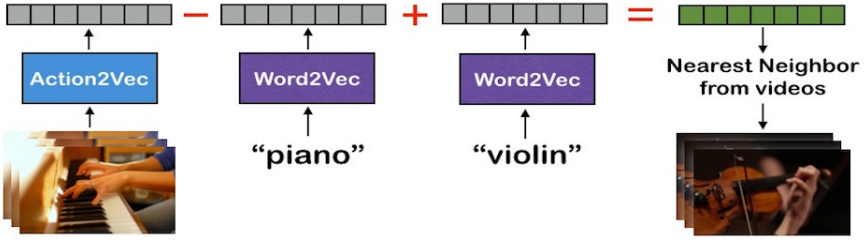

Figure 2: Vector Arithmetic Analogy Test: We take the Action2Vec descriptor for the video clip "play piano" and subtract the Word2Vec embedding of "piano" and add in "violin." The resulting vector matches the "play violin" video clip.

4.3.2 Vector Arithmetic

Here we describe a *novel analogy test for verbs* based on vector arithmetic, following previous analogy tests in distributional semantics and text-image embeddings [16]. Our test is designed to easily scale to most action datasets, in which multiple action classes contain the same verb with different nouns. For example, in Kinetics the verb "throw" has 8 instances ranging from throwing an ax to throwing a Frisbee. We perform the following vector arithmetic for each verb with multiple noun instances: $verb\ noun1 - noun1 + noun2 = V_{new}$. We then search the vector space of action classes and select the class whose vector is closest to V_{new} in euclidean distance. If $V_{new} = verb\ noun2$ we count this as correct. An example: $throw\ softball - softball + football = throw\ football$. As in §4.3.1, in order to get a single embedding for an action class, we average the Action2Vec vectors for all videos of that action class. We performed this test for Kinetics and UCF101 but not HMDB51, as it is too small and does not have multiple nouns per class. From Table 3 we can see that the vector arithmetic does not perform as well on Kinetics as it does on UCF101. We found that it failed on ambiguous actions such as "doing" and "making," which are missing from UCF101, making Kinetics more difficult.

5 Conclusion

The integration of vision and language is increasingly important for the advancement of AI. While joint text and image representations have been studied, the combination of text and video remains largely unexplored. Video is a natural modality for leveraging verb semantics, and the relationship between a verb and its video representation is complex. This paper introduces *Action2Vec*, a novel embedding space for actions which combines linguistic cues with spatio-temporal visual cues derived from deep video features generated by a hierarchal RNN model. We demonstrate that Action2Vec effectively captures discriminative visual features, and delivers state of the art zero shot action recognition performance. We also show that Action2Vec closely captures the distributional semantics of WordNet via its similarity

to the Word2Vec embedding. We propose two novel analogy tests for verb embeddings and use them to evaluate Action2Vec. We believe this is the first thorough evaluation of a video-text embedding space with respect to accuracy and semantics. We hope that Action2Vec can provide a useful intermediate representation for tasks in video generation, video retrieval, and video question answering.

6 Supplementary

6.1 Linguistic Elements of Verbs

After developing action embeddings based on verb semantics and demonstrating their quality by running analogy tests, we now describe some additional tools for analyzing the relationships between the action and verb manifolds. To do this, we leverage manually-curated lexicons that describe the syntactical groupings of verbs.

Verbs play a crucial role in the English language because they impose both semantic and structural constraints on a sentence. These constraint rules are defined by English syntax and grammar. In linguistics, these rules are often referred to as thematic roles [26]. Verbs are sorted into thematic roles based on how they act in relation to the other parts of speech in the sentence. For example, the verb "carry" needs an animate subject that will do the act of carrying, an object that can be carried from an initial location, and a destination. These arguments are the possible thematic roles for the verb "carry." The two most common arguments are the Agent and the Patient. The agent is the subject who carries out an action and the Patient is the object that receives the action. For example, in the sentence "Jon hit the car," Jon is the patient and the car is the patient. Some of the main thematic roles are as described [26]:

- **Agent**: The entity that carries out the action of the verb.
- **Patient**: The entity that undergoes a change of state because of the action.
- **Recipient**: The target of some transfer of possession which the verb indicates.
- **Beneficiary**: The entity that benefits from the action.
- **Instrument**: The entity through which the action is carried out.
- **Location**: Place introduced by a locative or path prepositional phrase.
- **Theme**: When participants in an action undergo a change in location.

Categorizing verbs based on their thematic roles is one way to examine actions in a semantic space. Examining groupings of verbs based on their thematic roles give insight into how they are used to affect the world without explicitly naming the things which they affect. The way verbs group in the semantic space of Word2Vec is actually quite different than the way verbs group by their thematic roles.

In Word2Vec the context of a word, w, is defined by the preceding and succeeding words around the original word. The model takes in the adjacent words $(w_{i-2}, w_{i-1}, w_{i+1} w_{i+2})$ and tries to predict the word w. Since Word2Vec is trying to predict the exact words around a verb, the resulting embedding is heavily based on object names rather than relational structures. Thematic roles create more generalized groupings of verbs based on their relational

structures. To understand the action embeddings more deeply, we compare way that Action2Vec embeddings cluster to the way verbs are clustered, both in the Word2Vec space and by their thematic roles. The Unified Verb Index is a lexicon that contains manually annotated thematic roles for almost all verbs in the English language. The Unified Verb Index is a combination of the lexicons: VerbNet [26], Framenet [3], and the Proposition Bank [15].

We use K-means to cluster the embeddings in both the action space and language space, and then compare the resulting clusters against each other and against the organization of verbs based on thematic roles which are derived from the Unified Verb Index. We show an example of the resulting thematic role clusters for the UCF101 dataset in Table 5. Through clustering, we gain additional insight into how the different types of embeddings are arranged in the manifolds that we have created. This gives insight into what information the embedding architectures are capturing.

6.2 Clustering and Lexical Analysis

For both the UCF101 and HMDB51 datasets we go through each class name and get the thematic roles for the verb from the Unified Verb Index. For classes that are phrases rather than a single verb, we use the thematic role of the verb portion of the phrase. For example, for the class "climb rope" we assign the thematic role of the verb "climb". We then cluster the the classes based on their thematic roles. For UCF101, these thematic-role clusters are shown in Table 5. For UCF101 we get 9 thematic-role clusters of verbs and for HMDB51 we get 7 thematic-role clusters of verbs. We then run K-means on the video embeddings and on the word embeddings for the class names, where we set K to 9 and 7 for the UCF101 embeddings and the HMDB51 embeddings, respectively. Then we use the Adjusted Rand Index (ARI) measurement [9] to evaluate the similarity between the Action2Vec embeddings in the action space, the Word2Vec embeddings in the semantic language space and the syntactic space of thematic roles. We use the ARI specifically to compute the similarity measure between the clusterings. The ARI computes a similarity between two clusters by calculating the percent of pairs of points that are shared between two clusters. The ARI similarity goes from 0 to 1, where 1 is perfect similarity and 0 is completely disjoint.

	Adj. Rand Index	
Cluster_1 vs. Cluster_2	HMDB51	UCF101
Action2Vec vs. Word2Vec	.1391	.4255
Action2Vec vs. thematic roles	.2025	.5442
Word2Vec vs. thematic roles	.1883	.3382

Table 4: The Action2Vec and Word2Vec embeddings for the classes of the datasets UCF101 and HMDB51 are clustered by K-means. The classes are also clustered based on class name's thematic roles as taken from the Unified Verb Index. We compare these clusters two at a time using the Adjusted Rand Index similarity measurement. For HMDB51, K-means is run with K=7 and for UCF101 K=9.

The ARI scores for the clusters from UCF101 and HMDB51 are shown in Table 4. As we can see in Table 4, for both datasets, the clusters in the Action2Vec space have a higher similarity to the thematic-role-based clusters than the clusters in the Action2Vec space have to the clusters in Word2Vec space. We reason this happens because while the word-embedding space is focused on embedding verbs based on information about the objects related to the

action, the thematic roles cluster verbs more generally and with greater regard for the motion and body parts used in the action. For example, "pizza tossing" has the thematic roles: $Agent, Theme, InitialLocation, Destination, Result$. These thematic roles are shared by all verbs in the class "throw." This is different from Word2Vec, where "throwing a discus" and "tossing a pizza" are not very similar because they don't revolve around similar objects or scenes. However in the action space, Action2Vec, these two videos are actually in the same cluster, because the motion and body parts of throwing are similar and the HRN is able to capture this similarity even in different scenes and with different objects.

Figure 3 shows the Action2Vec clusters. Using t-SNE, we plotted the way the actions cluster. We do the same thing for the Word2Vec embeddings, and visualize it in Figure 4. In the figures, we made clusters the same color for both embedding types based on the overlap between class names in the clusters. For instance, the orange cluster in both Figure 3 and Figure 4 contains most of the music-playing classes. We can see that the embedding spaces are quite different, based on the fact the clusters are in quite different positions on the graph. However, there are still clear groupings among the verbs which correspond to meaningful similarity among either the semantic name of the verb or the motion of the action. We can then compare these clusters to Table 5 which lists the verb clusters based on thematic roles.

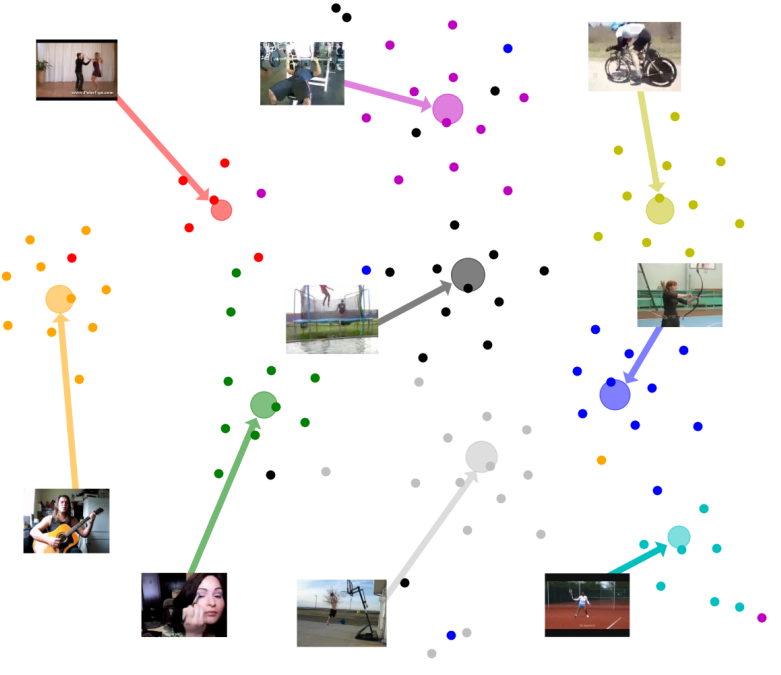

Figure 3: A t-SNE visualization of our 9 Action2Vec clusters and the mean vectors of each action in the cluster. Each point is the mean vector of one action class embedding, and the large points represent the mean of a cluster, scaled by the number of elements in the cluster. Best viewed in color.

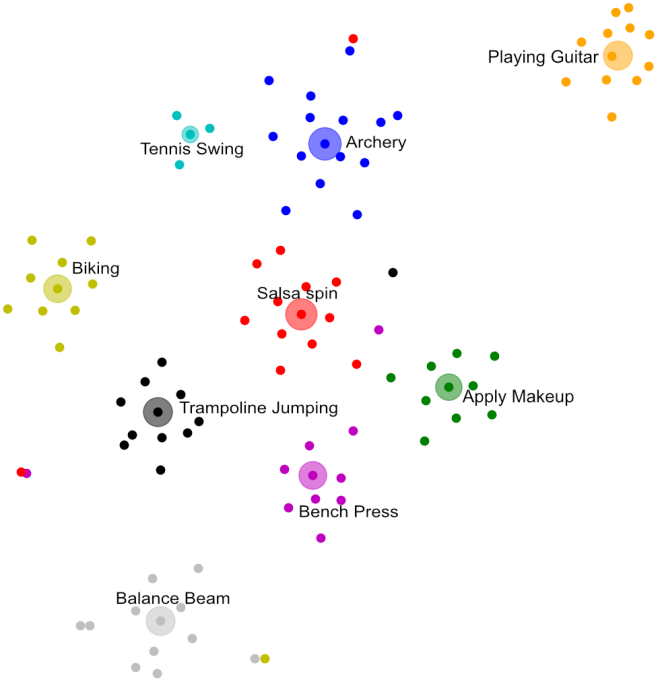

Figure 4: A t-SNE visualization of 9 Word2Vec clusters and the mean vectors of each action in the cluster. Each point is vector of one verb embedding, and the large points represent the mean of a cluster, scaled by the number of elements in the cluster. Best viewed in color.

Thematic Roles	UCF101 Classnames
Agent [animate, machine], Theme, Destination	Knit, Write_on_Board, Type, Blow_Candles
Agent [animate], Patient [concrete], Instrument [solid], Location, Result	Box_Punching_Bag, Box_Speed_Bag, Punch, Cut, Mix, Table_Tennis_Shot, Hammer, Fence, Mop_Floor, Nunchucks
Agent [int_control], Theme [concrete], Location, Result	Bowl, Field_Hockey_Penalty, Golf_Swing, Socce_Penalty, Swing, Tennis_Swing, Salsa_Spin, Hula_Hoop
Agent [animate], Location	BodyWeightSquats, HandstandPushups, HandstandWalking, Pullups, WallPushups, BenchPress, StillRings, UnevenBars, ParallelBars, Pushups, Lunges, CleanAndJerk, TaiChi, SumoWrestling
Agent [animate], Theme [animate], Location, Result	Bike, IceDance, Kayak, Raft, Row, Ski, Skijet, Surf, HorseRace, RideHorse, MilitaryParade, BreastStroke, FrontCrawl, WalkDog, BandMarch, BabyCrawl
Agent [animate], Theme [animate], Location	SkyDive, CliffDive, HighJump, JumpingJack, JumpRope, LongJump, RockClimbIndoor, TrampolineJump, RopeClimb, Dive, BalanceBeam, FloorGymnastics PoleVault, PommelHorse, SkateBoard, FrisbeeCatch
Agent [int_control], Theme [concrete], Initial_Location, Destination, Result	BasketballDunk, Basketball, HammerThrow, JuggleBalls, SoccerJuggle, VolleyballSpike, JavelinThrow, Shotput, ThrowDiscus, PizzaToss, YoYo, BaseballPitch, Archery, Billiards, CricketBowling, CricketShot
Agent [animate], Theme, Beneficiary	Drum, PlayCello, PlayDaf, PlayFlute, PlayGuitar, PlayGuitar, PlayPiano, PlaySitar,PlayTabla, PlayViolin, PlayDho

Table 5: Shows the clusters formed by clustering the UCF101 class names (in verb form) based on their thematic roles. The right column shows the verbs in each cluster. The left column lists the thematic roles for the particular cluster.

References

[1] Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *NAACL*, pages 19–27, 2009.

[2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and vqa. *arXiv preprint arXiv:1707.07998*, 2017.

[3] Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *ACL*, pages 86–90. Association for Computational Linguistics, 1998.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *arXiv preprint arXiv:1705.07750*, 2017.

[5] Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. *arXiv preprint arXiv:1803.08495*, 2018.

[6] Jianfeng Dong, Xirong Li, and Cees GM Snoek. Word2visualvec: Image and video to sentence matching by visual feature prediction. *CoRR*, 2016.

[7] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, pages 1933–1941, 2016.

[8] Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *NAACL Student Research Workshop*, pages 8–15, 2016.

[9] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[10] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, June 2015.

[11] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[12] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[13] Maurice G Kendall. Rank correlation methods. 1955.

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[15] Paul Kingsbury and Martha Palmer. From treebank to propbank. In *LREC*, pages 1989–1993, 2002.

[16] Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[17] Elyor Kodirov, Tao Xiang, Zhenyong Fu, and Shaogang Gong. Unsupervised domain adaptation for zero-shot learning. In *ICCV*, pages 2452–2460, 2015.

[18] Hilde Kuehne, Hueihan Jhuang, Rainer Stiefelhagen, and Thomas Serre. Hmdb51: A large video database for human motion recognition. In *High Performance Computing in Science and Engineering âĂŸ12*, pages 571–582. Springer, 2013.

[19] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[20] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[21] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 1995.

[22] Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *CVPR*, pages 1029–1038, 2016.

[23] Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, pages 4594–4602, 2016.

[24] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[25] Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marin, Ferda Ofli, Ingmar Weber, and Antonio Torralba. Learning cross-modal embeddings for cooking recipes and food images. *CVPR*, 2017.

[26] Karin Kipper Schuler. Verbnet: A broad-coverage, comprehensive verb lexicon. 2005.

[27] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014.

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

[29] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[30] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015.

[31] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, pages 3156–3164, 2015.

[32] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, June 2016.

[33] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.

[34] Xun Xu, Timothy Hospedales, and Shaogang Gong. Semantic embedding space for zero-shot action recognition. In *ICIP*, pages 63–67. IEEE, 2015.

[35] Xun Xu, Timothy M Hospedales, and Shaogang Gong. Multi-task zero-shot action recognition with prioritised data augmentation. In *ECCV*, pages 343–359. Springer, 2016.

[36] Xun Xu, Timothy Hospedales, and Shaogang Gong. Transductive zero-shot action recognition by word-vector embedding. *International Journal of Computer Vision*, pages 1–25, 2017.

[37] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *CVPR*, pages 4651–4659, 2016.