Semantic Aware Video Transcription Using Random Forest Classifiers

Chen Sun and Ram Nevatia

University of Southern California, Institute for Robotics and Intelligent Systems, Los Angeles, CA 90089, USA

Abstract. This paper focuses on transcription generation in the form of subject, verb, object (SVO) triplets for videos in the wild, given off-the-shelf visual concept detectors. This problem is challenging due to the availability of sentence only annotations, the unreliability of concept detectors, and the lack of training samples for many words. Facing these challenges, we propose a Semantic Aware Transcription (SAT) framework based on Random Forest classifiers. It takes concept detection results as input, and outputs a distribution of English words. SAT uses video, sentence pairs for training. It hierarchically learns node splits by grouping semantically similar words, measured by a continuous skip-gram language model. This not only addresses the sparsity of training samples per word, but also yields semantically reasonable errors during transcription. SAT provides a systematic way to measure the relatedness of a concept detector to real words, which helps us understand the relationship between current visual detectors and words in a semantic space. Experiments on a large video dataset with 1,970 clips and 85,550 sentences are used to demonstrate our idea.

Keywords: Video transcription, random forest, skim-gram language model.

1 Introduction

Humans can easily describe a video in terms of actors, actions and objects. It would be desirable to make this process automatic, so that users can retrieve semantically related videos using text queries, and capture the gist of a video before watching it. The goal of this paper is generating video transcriptions, where each transcription consists of a subject, verb and object (SVO) triplet. We assume the videos to be unconstrained user captured videos possibly with overlaid captions and camera motion, but that they are pre-segmented to be short clips with a single activity, and a few objects of interest. One example is shown in Figure 1 left.

Video transcription with SVO is an extremely challenging problem for several reasons: first, as annotating actions and objects with spatio-temporal bounding boxes is time-consuming and boring, in most cases, only video-level sentence annotations are available for training (Figure 1 right). Second, although there

D. Fleet et al. (Eds.): ECCV 2014, Part I, LNCS 8689, pp. 772–786, 2014.

Human annotations:
Three men are biking in the woods
Two cyclist do tricks
Guys are riding motorcycles
People ride their bikes
...

Output of SAT:
Person rides bike

Fig. 1. *Left:* one example of the testing videos we used. *Right:* our algorithm utilizes sentence based annotations, and output subject, verb, object triplets.

are several action and object datasets with a large number of categories [7,26], a considerable amount of SVO terms are still not present in these categories. Finally, even for the detectors with corresponding SVO terms, many of them are still far from reliable when applied to videos in the wild [15].

Many papers on activity analysis have emerged recently. Usual goal is activity or event classification of pre-defined categories [32,27,18]. For video transcription problem where the combinatorial space of SVO triplets is much bigger and sparse, it is hard to apply these techniques directly and learn a classifier for every SVO triplet. Guadarrama *et al.* [14] proposed a video transcription method YouTube2Text: they learned an SVM classifier for each term in candidate subjects, verbs and objects, and used low-level and mid-level visual features [17] [30] for classification. All these approaches treat each class (either an *activity* or a *term*) independently, and ignore the semantic relationships between the classes.

We propose a semantic aware transcription framework (SAT) using Random Forest classifiers. Inputs for the Random Forest classifiers are detection responses from off-the-shelf action and object detectors. SAT's outputs are in the form of SVO triplets, which can then be used for sentence generation. To obtain the SVO terms for training, we parse human annotated sentences and retrieve the subject, verb and object terms. The labels of a training video contain the top k most commonly used subject, verb and object terms for that video. For example, the set of labels for video in Figure 1 may be (*person, motorcyclist, ride, do, bicycle, trick*).

The core innovation of SAT is to consider the semantic relationships of SVO labels during training. Semantic aware training is important when the labels are user provided without a pre-defined small vocabulary set. On one hand, humans may use different words to describe objects that are close visually or essentially the same (*bike* and *bicycle*). On the other hand, for problems with a large number of classes, semantically reasonable errors (*tomato* to *potato*) are more desirable than unreasonable ones (*tomato* to *guitar*). SAT provides a framework for semantic aware training: during node split selection of decision trees, it favors the clustering of semantically similar words. Similarity is measured by continuous word vectors, learned with the skip-gram model [21]. The skip-gram model optimizes context prediction power of words over a training corpus, and has been

shown to produce word vector clusters with clear semantic similarities. Given the learned word vectors, SAT picks the best node split by computing differential entropy of word clusters. Each tree in the resulting forest divides training samples hierarchically into semantically consistent clusters.

The detector responses used in this paper can be seen as candidate action and object proposals. They are more suitable for the transcription task than low level features, as action and object locations are not provided in the annotations. Torresani *et al.* [29] showed that object detector responses provide competitive performance when used as features for image classification task. SAT goes one step further and provides a mechanism to measure the semantic map from a detector type to output labels. The map measures the influence of a detector's response on the output probabilities of labels. For example, *bicycle* detector may have high impact on both objects like *bike* or *motorcycle* as well as verbs like *ride*.

SAT has the following highlights:

Larger vocabulary support. A Random Forest classifier is naturally suited for multi-class classification. We can use a single Random Forest for arbitrary vocabulary size. For SVM-based frameworks, the number of one-vs-rest classifiers required grows linearly with vocabulary size.

Feature sharing for semantically similar words. By using a hierarchical structure, SAT allows sharing features for semantically similar words. For example, *horse* and *bicycle* may go through the same path in a decision tree until separated by a node with large tree depth. This is particularly useful for training as words with few occurrence can be trained together with similar words with more training samples.

Semantic reasonableness. SAT optimizes over semantic similarity instead of binary classification error. In our framework, *piano* is considered a *better* error for *guitar* than *pasta*. The resulting transcriptions are thus likely to be more semantically reasonable.

The contribution of this paper is two-fold: First, we propose a semantic aware learning algorithm for Random Forest classifiers, which has the potential of producing semantically reasonable results. Second, we provide a mechanism to compute semantic maps from detectors to words.

2 Related Work

Several recent papers have focused on generating descriptions for visual contents. Kulkarni *et al.* [16] proposed a method to detect candidate objects and their attributes from static images, and applied CRF for sentence generation. [1] used object detection and tracking results to describe videos with simple actions and fixed camera. In [15], the authors obtained the SVO triplets using object and action detectors and reranked the triplet proposals with language models. A related task to video description is event recounting, it asks a system to output supporting evidence for a video event. [6,19] used event labels as prior and built CRF or SVM models with concept detector responses. All these approaches

assume that the detectors carry direct semantic meanings and require trained detectors for every action, object or attribute of interest. SAT is different from these as it learns a hierarchical mapping from detector response space to word space.

Alternatively, [25] proposed to classify semantic representations (SR) with low level features, and used a CRF to model the co-occurrences of SR. It formulated the conversion from SR to sentences as a statistical machine translation problem and tested the idea on an indoor kitchen dataset. However, global low level features may not be discriminative enough to identify actions and objects for videos in the wild.

The idea of utilizing semantic relationships of annotations have motivated several papers on image and video analysis. Topic model was used in [24] to convert text into topic distribution vectors and group mid-level actions. Deng *et al.* [8] observed the existence of a trade-off between accuracy and specificity for object categories and applied it to image classification. Specificity is measured by an object's depth in WordNet hierarchy. YouTube2Text system [14] extended this idea and used data-driven hierarchies to generate SVO video transcriptions. Both of them applied semantic hierarchy in the post-processing stage. Unlike these approaches, SAT uses word vectors in a continuous semantic space, and defines an adaptive similarity measurement for arbitrary word sets; semantic similarity is explicitly used to learn the word maps.

3 Proposed Method

This section describes the Semantic Aware Transcription framework. We first briefly introduce a vector based word representation in semantic space [21]. The structure of Random Forest classifiers and their inputs are then described. Next, we show how the semantic word vectors can be used to select the best node split in Random Forest classifier training, such that training samples after split become more similar in the semantic space. Finally, a mechanism is provided to compute the semantic map for a concept detector.

3.1 Continuous Word Representation

Many existing Natural Language Processing techniques can be used to measure semantic distances among different words. For example, WordNet [23] provides a database of hierarchical word trees, on which semantic distances can be defined. To learn data driven semantic structures, topic modeling techniques such as Latent Dirichlet Allocation have been found to be useful [3].

We adopt the continuous word representation learned by skip-gram model [22]. Given a sequence of training words $\{w_1, w_2, ..., w_T\}$, it searches for a vector representation for each word w_i, denoted by v_{w_i}, such that

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log P(w_{t+j}|w_t) \tag{1}$$

is maximized. c controls the training context size, and the probability of w_{t+j} given w_t is defined by the softmax function

$$P(w_i|w_j) = \frac{\exp(v_{w_i}^T \ v_{w_j})}{\sum_w \exp(v_w^T \ v_{w_j})} \tag{2}$$

This objective function attempts to make the vector representation of semantically close words behave similarly in predicting their contexts. In practice, a hierarchical softmax function is used to make the training process computationally feasible. When trained on large text corpus, the Euclidean distances between vectors of semantically similar words are small.

Compared with rule-based WordNet and the topic modeling techniques, continuous word representation is both data-driven and flexible. Once word vectors are trained from an independent corpus, one can measure the semantic similarity for an arbitrary set of words.

3.2 Video and Annotation Preprocessing

We assume each training video has several one-sentence descriptions annotated via crowdsourcing. These sentences are parsed by a dependency parser [20], and only subject, verb and object components are kept. Denote D_s, D_v and D_o as the dictionary of subjects, verbs and objects, we store their word vectors as $V_s = \{v_{w_s}|w_s \in D_s\}$, $V_v = \{v_{w_v}|w_v \in D_v\}$, $V_o = \{v_{w_o}|w_o \in D_o\}$.

After annotation preprocessing, every training video has a set of SVO words. For subject and object words, although most of them correspond to concrete objects, we lack the bounding boxes to locate them. Meanwhile, an annotated verb may correspond to very different actions, like the verb *play* in *play guitar* and *play soccer*. It is hard to learn verb detectors based on these annotations directly.

We use off-the-shelf action and object detectors to represent a video [13,28]. Training data for these detectors are obtained from independent datasets. The types of trained detectors correspond to a very limited vocabulary and may not contain the words used in video transcriptions. To apply object detectors, we sample video frames and take the maximum response returned by a detector over all sampled frames; action detectors are applied with sliding windows, and are also combined by maximum pooling. The final video representation is a vector of action and object detector responses $S = [s_1 \ s_2 \ ... \ s_M]$. Each dimension corresponds to a type of action or object detector.

3.3 Random Forest Structure

As illustrated in Figure 2, we use a forest of randomly trained decision trees to map detector responses into posterior word probabilities.

Starting from the root, every non-leaf node k contains a simple classifier $\Phi_k(S)$ for vector S of detector responses based on a single type of detector response.

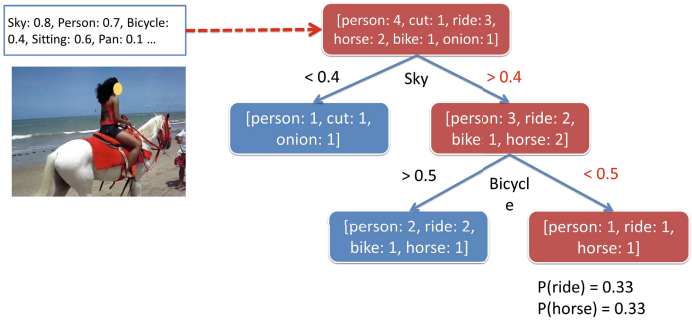

Fig. 2. Illustration of a single decision tree used in SAT. Detector responses for a video are used to traverse the tree nodes until reaching a leaf. Note that a horse detector may not be needed in the process.

We have

$$\Phi_k(S) = s_i - \tau \begin{cases} > 0 & \text{Go to left} \\ < 0 & \text{Go to right} \end{cases} \tag{3}$$

where s_i is the i-th concept in the vector, and τ is the threshold.

Leaf nodes store word count vectors; as in traditional decision trees, a word count vector is obtained by accumulating the SVO words from all training samples belonging to the leaf node. The final confidence score for word w_i is obtained by

$$f(w_i) = \frac{1}{T} \sum_{t=1}^{T} \frac{c_{t,w_i}}{\sum_w c_{t,w}} \tag{4}$$

where T is the forest size, $c_{t,w}$ is the count for word w at the leaf node of the t-th decision tree.

The subject, verb and object terms with the highest confidence scores respectively are selected to generate a sentence description for a video.

3.4 Learning Semantic Hierarchies

Ideally, we would like to learn a tree structure which encodes the semantic hierarchy of SVO words. Towards this goal, we use the continuous word vectors to measure the *semantic compactness* for a set of words.

Denote $W = \{w_1, w_2, ..., w_M\}$ as a group of words, and $V = \{v_{w_1}, v_{w_2}, ..., v_{w_M}\}$ the word vectors. Assume the underlying distribution of the word vectors is Gaussian, we have

$$g(v_w) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp\left(-\frac{1}{2}(v_w - \mu)^T \Sigma^{-1} (v_w - \mu)\right) \tag{5}$$

where k is the dimension of word vectors, $\mu = [\mu_1\ \mu_2\ ...\ \mu_k]$ is the mean vector, and $\Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_k)$ is the diagonal covariance matrix. They can be estimated from V by

$$\mu_j = \frac{1}{M}\sum_{i=1}^{M} v_{w_i}^j \tag{6}$$

$$\sigma_j = \frac{1}{M}\sum_{i=1}^{M}(v_{w_i}^j - \mu_j)^2 \tag{7}$$

In analogy to entropy defined on discrete variables, we compute differential entropy $H(\mu, \Sigma)$ for the Gaussian distribution parametrized by Σ and μ following

$$H(\mu, \Sigma) = \frac{1}{2}\ln|(2\pi e)\Sigma| \sim \sum_{j=1}^{k} \ln \sigma_j + C \tag{8}$$

$H(\mu, \Sigma)$ measures the degree of uncertainty for the distribution: the lower the value, the more certain the distribution is. For word vectors, since semantically similar words lie close to each other, their estimated σ's should be small and the differential entropy low according to Equation 8. As a result, to achieve semantic compact node splits, we minimize the weighted differential entropy

$$\frac{|V_l|}{|V_l| + |V_r|} H(\mu_l, \Sigma_l) + \frac{|V_r|}{|V_l| + |V_r|} H(\mu_r, \Sigma_r) \tag{9}$$

where V_l and V_r are the two groups of word vectors after node split.

It has been shown that the generalization error of random forests is determined by the strength of individual trees and correlation of the trees [4]. To reduce correlation, we impose several types of randomness in training. First, only a subset of training videos is sampled to train each decision forest. Second, we randomly assign each node to consider only subject words, verb words or object words, and use the selected word to compute differential entropy as defined in Equation 8. Finally, we use the node split selection criteria similar to extremely randomized trees [12]: after a feature dimension is sampled, instead of finding the best threshold to minimize Equation 9, we only choose a small subset of candidate thresholds and pick the best one among them. The training algorithm is summarized in Algorithm 1.

3.5 Computing Semantic Maps

In Section 3.2, we showed how to obtain a video representation based on pre-trained action and object detectors. SAT does not require a detector to carry direct semantic meaning indicated by its name, but uses its response to traverse the semantic hierarchy defined by Random Forest classifiers.

Motivated by the variable importance estimation for Random Forest classifiers [4], we use a similar scheme to compute semantic maps from input detector types to output words. Let M be the number of action and object detectors, and F be the trained Random Forest classifier. Given a set $\mathcal{S}$ of detector response vectors, for each detector type m, we set its value for all vectors in $\mathcal{S}$ to the lowest possible and the highest possible, and fix the values of all the other dimensions.

Algorithm 1. SAT Training Algorithm

Input: A set $\mathcal{S}$ of training videos as concept response and word annotation pairs
Output: Random forest with T decision trees
for $t = 1$ to T **do**
  Sample a subset of $\mathcal{S}$ as $\mathcal{S}_t$
  Call **splitNode**($\mathcal{S}_t$)
end for

splitNode($\mathcal{S}$):
if stop criteria not met **then**
  Randomly select a node type from SVO
  Randomly sample N_f feature dimension indices and N_t thresholds
  Apply each weak classifier to split set $\mathcal{S}$
  Evaluate weak classifiers using words of selected type (Equation 9)
  Select the weak classifier which minimizes Equation 9
  Split $\mathcal{S}$ into $\mathcal{S}_l$ and $\mathcal{S}_r$ based on the selected weak classifier
  Call **splitNode**($\mathcal{S}_l$) and **splitNode**($\mathcal{S}_r$)
else
  Compute word counts and mark the node as leaf
end if

This produces two modified sets $\mathcal{S}'$ and $\mathcal{S}''$. We apply F on $\mathcal{S}'$ and $\mathcal{S}''$ to get the word probabilities. The correlation of the m-th detector and a word w is measure by

$$\varphi_m(w) = \sum_{i=1}^{N} |f'_w(i) - f''_w(i)| \tag{10}$$

Here $f'_w(i)$ is w's probability for the i-th sample in $\mathcal{S}'$, and $f''_w(i)$ is its probability for the i-th sample in $\mathcal{S}''$. Higher value of $\varphi_m(w)$ indicates a strong map between the detector type and the word.

3.6 Discussion

One major difference of SAT from traditional Random Forest classifiers is the node split criteria during training. SAT fits a group of semantic word vectors using a Gaussian distribution with diagonal covariance matrix, and computes differential entropy to measure the *semantic compactness*. The penalties of grouping semantic similar words are smaller. For example, the split of (*drive, ride*) and (*cut, slice*) should be better than (*drive, slice*) and (*cut, ride*). Traditional Random Forest classifiers cannot distinguish the two as their discrete entropies are the same. This difference makes SAT produce more semantically reasonable predictions.

Video transcription using SAT is fast (tens of comparisons for each decision tree, and hundreds of trees in total). For training, it only evaluates the randomly sampled thresholds instead of searching for the optimum, which can be done very

efficiently. Since there is no interaction between different trees, both training and testing of SAT can be parallelized easily.

Our method to compute semantic map is related to, but different from, variable importance estimation: we measure only the change in output word probabilities; instead of filling in randomly selected values, we select only the maximum and minimum possible values for that dimension, so that all nodes using this dimension to make decision are toggled.

Computed semantic maps provide several indications: if the semantic meaning of a detector's name and its top mapped words are identical or very similar, it is quite likely that the detector outputs are reliable. Besides, if an object detector's top mapped words contain verbs or an action detector's top mapped words contain objects, the combination should appear frequently in training videos.

4 Experiments

In this section, we first describe our experiment setup and the dataset used for evaluations. Next, we compare the performance of SAT with several other video transcription frameworks. Semantic maps learned by SAT are shown at the end of this section.

4.1 Dataset

We used the YouTube dataset collected by Chen and Dolan [5]. There are 1,970 short video clips with 85,550 sentence-based video descriptions in English. Videos were annotated by Amazon Mechanical Turk workers.

Object detector training data were provided by PASCAL VOC challenge [9] and a subset of ImageNet [7]. There are 243 categories in total. For action detector training, we used UCF 101 dataset [26] with 101 categories.

4.2 Experimental Setup

We followed data partitioning used in [14], there are 1,300 training videos and 670 testing videos.

Stanford parser [20] was used to extract the subject, verb and object components from the sentence associated with the videos. Some of the extracted words are typos or occur only a few times, we filtered these words out; this results in a dictionary size of 517 words. As annotators tend to describe the same video with diverse words, each video was described by the one most common subject, the two most common verbs and the two most common objects. Unless otherwise specified, we used this set of words as the groundtruth to train the classifiers and measure the accuracy of video transcription.

We used the continuous word vectors pre-trained on Google News dataset. It was provided by the authors of [22]. The dimension of each word vector is 300.

Deformable part models (DPM) [11] were used to train object detectors. Part of the detector models were downloaded from [2]. The object detector works on

static frames. We uniformly sampled frames every second, and used maximum pooling to merge the detector confidence scores for all sampled frames in the same video.

To learn action detectors, we first extracted motion compensated dense trajectory features with default parameters [31], and encoded the features with Fisher Vectors [28]. We set the number of clusters for Fisher Vectors as 512, and computed Fisher Vectors for the HOG, HOF and MBH components separately. A linear SVM [10] was then trained for each action category with a single type of features. We used average fusion to combine the classifier outputs.

Parameter set for Random Forest classifiers includes the number of decision trees T, the number of sampled feature dimensions N_f and thresholds N_t, as well as the max tree depth D. Parameters were selected by measuring out-of-bag errors (OOB) [4]. It was computed as the average of prediction errors for each decision tree, using the non-selected training data.

Table 1. Top correlated verb and object pairs in SAT

Verb	Top correlation
come	go
run	walk
spread	mix
fry	cook
put	pour

Object	Top correlation
scooter	bicycle
finger	hand
motorbike	car
vegetable	onion
computer	camera

4.3 Performance Evaluation

We first qualitatively show how SAT uses semantic similarity to group training samples. We computed correlation of two words based on their number of cooccurrences in SAT's leaf nodes. To avoid correlation introduced by multiple annotations for the same video, we used only a single SVO triplet for each video. Table 1 shows several subject or object words with their top correlated words. As we can see, most of the pairs are both semantically close and visually related.

For quantitative evaluation, we compare our proposed SAT framework with the following two baselines:

Random Forest with no semantic grouping (RF). Every word under this setting was treated as an independent class. Node split is selected by computing the discrete entropy.

Linear SVM (SVM). A linear SVM classifier was learned for every word, using the detector responses as input features.

We fixed $T = 150$ and $D = 40$ for SAT and RF. N_f and N_t were selected by OOB. For SVM system, we fixed the soft-margin penalty ratio between positive and negative samples as the inverse of their sample size ratio, and used cross validation to select the cost parameter.

Table 2. Accuracy comparison among our proposed SAT, a traditional RF and a linear SVM

Method	Subject accuracy	Verb accuracy	Object accuracy
SAT	**0.816**	**0.344**	**0.244**
RF	**0.816**	0.312	0.152
SVM	0.726	0.281	0.191

Table 2 shows the accuracy comparison for the three methods. It is easy to see that our proposed SAT provides better performance in both verb accuracy and object accuracy, compared with the other two systems which do not use semantic relationships during training. In Figure 3, we also show some of the transcription results. SAT provided the correct SVO triplets for the top two examples, and related triplets for the middle two examples. The bottom one is a case where SAT returned wrong result.

Table 3. Accuracy and WUP comparisons between our proposed method and YouTube2text [14]

Method	Subject accuracy	Verb accuracy	Object accuracy
SAT	0.792	**0.306**	**0.188**
YouTube2Text [14]	**0.809**	0.291	0.170
Method	Subject WUP	Verb WUP	Object WUP
SAT	**0.927**	**0.625**	**0.590**
YouTube2Text [14]	0.926	0.468	0.467

We also compare the performance of SAT with the YouTube2Text system proposed by [14]. It used semantic hierarchies to convert unconfident SVO proposals to terms with higher semantic hierarchy. Their evaluations included a binary accuracy measurement using only the most common SVO triplet per testing video, no semantic conversion was used for this evaluation. To make our results comparable, we used the groundtruth labels provided by the authors. WUP metric was also used for evaluation. It is computed by

$$s_{WUP}(w_1, w_2) = \frac{2 \cdot D_{lcs}}{D_{w_1} + D_{w_2}} \tag{11}$$

where lcs is the least common ancestor of w_1 and w_2 in the semantic tree defined by WordNet, and D_w is the depth of w in the semantic tree. It provides the semantic similarity of w_1 and w_2 defined by the rule-based WordNet. Since a word may have multiple entries in WordNet, we used the set of entries provided by [14].

In Table 3, the binary accuracy of SAT is comparable to YouTube2Text in subject terms, and better in verb and object terms. For the WUP measure where semantic relatedness is being considered, SAT outperforms the YouTube2Text system by a large margin.

GT: Person rides bicycle.
SAT: Person rides bicycle.
RF: Person tries ball.
SVM: Person rides bicycle.

GT: Person dances rain.
SAT: Person dances group.
RF: Person does hair.
SVM: Person kicks video.

GT: Person does exercise.
SAT: Person does exercise.
RF: Person does pistol.
SVM: Person gets pencil.

GT: Person runs ball.
SAT: Person plays ball.
RF: Person hits ball.
SVM: Person kicks garden.

GT: Person eats pizza.
SAT: Person makes food.
RF: Person goes something.
SVM: Person makes box.

GT: Person drives car.
SAT: Person rides car.
RF: Person moves bicycle.
SVM: Person does pool.

Fig. 3. Testing videos with SVO triplets from groundtruth (GT), SAT, RF and SVM. Exact matches are marked in blue, semantic related verbs and objects are marked in red.

Fig. 4. Visualization of semantic maps for some action and object detectors

4.4 Visualization of Semantic Maps

Finally, we visualize some of the detector semantic maps in Figure 4. The maps were computed using testing videos. It can be seen that some of the detectors have semantically close mappings in the word space. Many action detectors we used involve objects, this is reflected by their top mapped words (*board* for *drawing on board* detector). It is also interesting to see that some detectors are connected with the top mapped words through motion patterns (the word *dance* and the *salsa spin* detector). This observation also holds for object detectors. The maps also illustrate how SAT handles the words outside the detectors' vocabulary.

5 Conclusion

We propose a Semantic Aware Video Transcription (SAT) system using Random Forest classifiers. SAT builds a hierarchical structure using the response of action and object detectors. It favors grouping of semantically similar words, and outputs the probabilities of subject, verb, object terms. SAT supports large vocabulary of output words, and is able to generate more semantic reasonable results. Experimental results on a web video dataset of 1970 videos and 85,550 sentences showed that SAT provides state-of-the-art transcription performance.

Acknowledgement. This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via the Department of Interior National Business Center (DoI/NBC), contract number D11PC0067. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

References

1. Barbu, A., Bridge, A., Burchill, Z., Coroian, D., Dickinson, S.J., Fidler, S., Michaux, A., Mussman, S., Narayanaswamy, S., Salvi, D., Schmidt, L., Shangguan, J., Siskind, J.M., Waggoner, J.W., Wang, S., Wei, J., Yin, Y., Zhang, Z.: Video in sentences out. In: UAI (2012)
2. Batra, D., Agrawal, H., Banik, P., Chavali, N., Alfadda, A.: Cloudcv: Large-scale distributed computer vision as a cloud service (2013)
3. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. JMLR (2003)
4. Breiman, L.: Random forests. Machine Learning (2001)
5. Chen, D., Dolan, W.B.: Collecting highly parallel data for paraphrase evaluation. In: ACL (2011)
6. Das, P., Xu, C., Doell, R.F., Corso, J.J.: A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In: CVPR (2013)
7. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR (2009)
8. Deng, J., Krause, J., Berg, A., Fei-Fei, L.: Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition. In: CVPR (2012)

9. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV (2010)
10. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research (2008)
11. Felzenszwalb, P.F., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. PAMI (2009)
12. Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine Learning (2006)
13. Girshick, R.B., Felzenszwalb, P.F., McAllester, D.: Discriminatively trained deformable part models, release 5
14. Guadarrama, S., Krishnamoorthy, N., Malkarnenkar, G., Mooney, R., Darrell, T., Saenko, K.: Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In: ICCV (2013)
15. Krishnamoorthy, N., Malkarnenkar, G., Mooney, R.J., Saenko, K., Guadarrama, S.: Generating natural-language video descriptions using text-mined knowledge. In: AAAI (2013)
16. Kulkarni, G., Premraj, V., Dhar, S., Li, S., Choi, Y., Berg, A.C., Berg, T.L.: Baby talk: Understanding and generating image descriptions. In: CVPR (2011)
17. Li, L.J., Su, H., Xing, E.P., Li, F.F.: Object bank: A high-level image representation for scene classification & semantic feature sparsification. In: NIPS (2010)
18. Li, W., Yu, Q., Divakaran, A., Vasconcelos, N.: Dynamic pooling for complex event recognition. In: ICCV (2013)
19. Liu, J., Yu, Q., Javed, O., Ali, S., Tamrakar, A., Divakaran, A., Cheng, H., Sawhney, H.S.: Video event recognition using concept attributes. In: WACV (2013)
20. de Marneffe, M.C., MacCartney, B., Manning, C.D.: Generating typed dependency parses from phrase structure parses. In: LREC (2006)
21. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR (2013)
22. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS (2013)
23. Miller, G.A.: Wordnet: A lexical database for English. CACM (1995)
24. Ramanathan, V., Liang, P., Fei-Fei, L.: Video event understanding using natural language descriptions. In: ICCV (2013)
25. Rohrbach, M., Qiu, W., Titov, I., Thater, S., Pinkal, M., Schiele, B.: Translating video content to natural language descriptions. In: ICCV (2013)
26. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. CRCV-TR-12-01
27. Sun, C., Nevatia, R.: Active: Activity concept transitions in video event classification. In: ICCV (2013)
28. Sun, C., Nevatia, R.: Large-scale web video event classification by use of fisher vectors. In: WACV (2013)
29. Torresani, L., Szummer, M., Fitzgibbon, A.: Efficient object category recognition using classemes. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 776–789. Springer, Heidelberg (2010)
30. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: CVPR (2011)
31. Wang, H., Schmid, C.: Action Recognition with Improved Trajectories. In: ICCV (2013)
32. Wang, L., Qiao, Y., Tang, X.: Mining motion atoms and phrases for complex action recognition. In: ICCV (2013)