ECAI 2012
Luc De Raedt et al. (Eds.)
© 2012 The Author(s).
This article is published online with Open Access by IOS Press and distributed under the terms
of the Creative Commons Attribution Non-Commercial License.
doi:10.3233/978-1-61499-098-7-600

Improving Video Activity Recognition using Object Recognition and Text Mining

Tanvi S. Motwani [1] and **Raymond J. Mooney** [2]

Abstract. Recognizing activities in real-world videos is a challenging AI problem. We present a novel combination of standard activity classification, object recognition, and text mining to learn effective activity recognizers without ever explicitly labeling training videos. We cluster verbs used to describe videos to automatically discover classes of activities and produce a labeled training set. This labeled data is then used to train an activity classifier based on spatio-temporal features. Next, text mining is employed to learn the correlations between these verbs and related objects. This knowledge is then used together with the outputs of an off-the-shelf object recognizer and the trained activity classifier to produce an improved activity recognizer. Experiments on a corpus of YouTube videos demonstrate the effectiveness of the overall approach.

1 Introduction

Recognizing activities in real-world videos is a challenging AI problem with many practical applications [1, 16]. We present a novel approach to efficiently constructing activity recognizers by effectively combining three diverse techniques. First, we use natural-language descriptions of videos as "weak" supervision for training activity recognizers [15]. We automatically develop a set of activities together with a labeled training corpus by clustering the verbs used in sentences describing videos. Second, we use previously trained object recognizers to automatically detect objects in video and use this information to help identify related activities [14]. For example, detecting a "horse" in the image helps classify the activity as "riding". Third, we mine a large corpus of generic, raw natural-language text to learn the correlations between activities (verbs) and their related objects (nouns). By mining a large corpus and collecting statistics on how likely different verbs co-occur with particular nouns, we estimate the probability of specific activities given particular objects.

Integrating these three methods allows for the rapid development of fairly accurate activity recognizers without ever explicitly providing training labels for videos. By combining text mining to both automatically infer labeled activities and extract relevant world-knowledge connecting activities and objects, together with computer-vision techniques for both object and activity recognition, our work demonstrates the utility of integrating methods in natural language processing and computer vision to develop effective AI systems. Experiments on a sizeable corpus of YouTube videos annotated with natural-language descriptions [3] verify that our approach improves the accuracy of a standard activity recognizer for real-world videos. Figure 1 shows sample frames from a couple of videos with their linguistic descriptions.

The remainder of the paper is organized as follows. Section 2 discusses related work, Section 3 describes our new system, Section 4 experimentally evaluates it on real-world videos, Section 5 discusses future work and Section 6 presents our conclusions.

Figure 1. Sample Videos with Natural-Language Descriptions

2 Related Work

Video activity recognition has become an active area of research in recent years [8, 32]. However, the set of activity classes are always explicitly provided, whereas we automatically discover the set of activities from textual descriptions. Scene context [26] and object context [14, 27, 31, 35] has previously been used to aid activity recognition. But most of this previous work uses a very constrained set of activities, while we address a diverse set of activities in real-world YouTube videos.

Also, unlike previous work, we automatically extract correlations between activities and objects from a large text corpus. There has been work using text associated with videos in the form of scripts or closed captions to aid activity recognition [10, 20, 19, 4, 15]. However, these methods do not use deeper natural language processing. By contrast, we demonstrate the advantage of full parsing of an unrelated corpus to mine general knowledge connecting objects and activities.

A particular related project that uses natural-language descriptions to automatically annotate videos with activity labels is [19]. However, in [19], the set of activity classes are pre-specified, whereas we automatically generate activity classes from textual descriptions by clustering verbs using WordNet as the only source of prior knowledge or supervision. Also, the approach in [19] requires a training set in which linguistic descriptions are annotated with the pre-specified

[1] Department of Computer Science, The University of Texas at Austin, 1616 Guadalupe, Suite 2.408, Austin, TX 78701, USA, email: tanvi@cs.utexas.edu
[2] Department of Computer Science, The University of Texas at Austin, 1616 Guadalupe, Suite 2.408, Austin, TX 78701, USA, email: mooney@cs.utexas.edu

activities, whereas our approach does not require any specially annotated text. While our current approach uses training videos each described by several different natural-language sentences provided by multiple human annotators recruited on Amazon Mechanical Turk, [19] uses freely available movie scripts downloaded from the web.

3 System Description

We first describe our procedure for discovering activity classes automatically from natural-language descriptions of videos using hierarchical clustering of verbs. Videos are then automatically labeled with these classes based on the verbs used in their natural-language descriptions. We then explain the process of training an activity classifier using spatio-temporal video features and how this model is used to obtain initial probability distributions over activities in test videos. Next, we describe how we detected objects in the videos using an off-the-shelf object recognizer. After that, we describe how we mined text to determine the correlation between these activities and objects. Finally, we explain how we combined all of these pieces together to produce a final integrated activity recognizer.

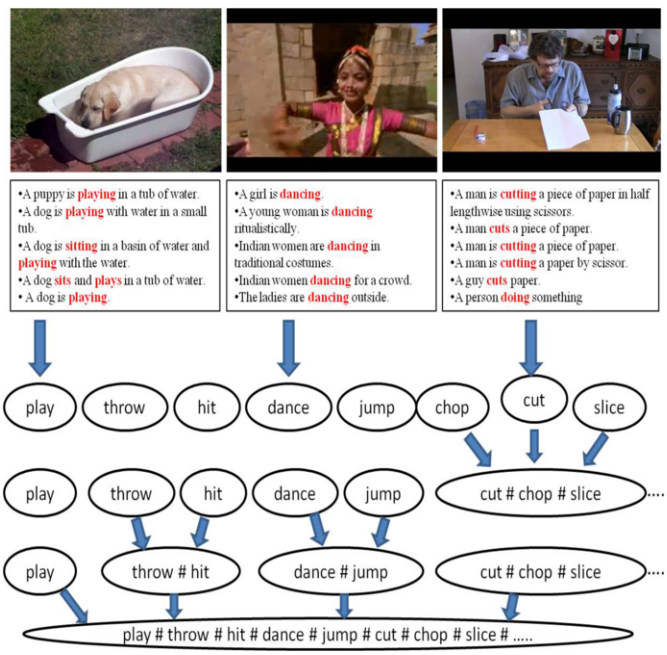

Figure 2. Automatic Discovery of Activity Classes

3.1 Automatically Discovering Activities and Producing Labeled Training Data

We assume that each training video is accompanied by a set of descriptive natural-language sentences and that there is one activity per video. These descriptions are used to automatically discover activity labels and produce labeled training data. First, we run the Stanford Log-Linear Part of Speech (POS) Tagger [33] on the sentences and compute the most frequent verb used to describe each video. Verbs are first stemmed using the Porter Stemmer [24]. Using the most-frequent verb in a set of natural-language descriptions of the video gives us high confidence that the activity is present in the video. These verbs could be used as activity labels themselves; however, this would result in a large set of activities, many of which are semantically similar. Thus, we automatically cluster these verbs to produce a smaller set of more distinctive activities.

We use WordNet::Similarity [28] to construct a semantic similarity measure between words. Specifically, word similarity is computed by summing three measures based on path lengths in WordNet: *lch* [21], *wup* [36] and *path*, and three others based on additional WordNet content: *res* [30], *lin* [23] and *jcn* [17]. When computing these similarity measures, we need the WordNet senses of the two words being compared. We tried two methods for measuring the final similarity of two words with unknown senses. In the first, we summed the similarity measures for each possible combination of the senses of the two words; in the second, we measured the similarity between the most-common senses of the two words according to WordNet. In the future, we would like to perform *word sense disambiguation* for these verbs using their textual as well as visual context. These similarity measures were used to produce complete binary taxonomic hierarchies of verbs by applying standard *Hierarchical Agglomerative Clustering* using group average to compare clusters [25].

For our dataset, we found that the approach using most-common senses worked best. We cut the resulting hierarchy at a level that seemed to provide the most meaningful activity labels and discarded clusters having fewer than 9 training videos. Automating the selection of the appropriate number of clusters is another topic for future work. Figure 2 shows how the similar verbs "cut," "chop" and "slice" were clustered together, similarly for "throw" and "hit" etc. Figure 3 shows the final 28 clusters discovered for our data. Finally, we automatically label each training video with the label of the cluster containing its original verb label.

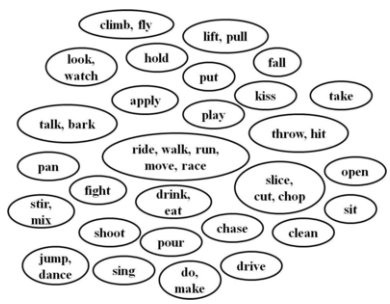

Figure 3. Discovered Verb Clusters that Define Activities

3.2 Activity Classification using Spatio-Temporal Video Features

The labeled videos produced in the previous step are used to train a standard video-based activity classifier. First, each video clip is preprocessed to produce a descriptive "bag of visual words." To capture spatially and temporally interesting movements, we use the motion descriptors developed by Laptev [18]. These features have been shown to work well for human-activity recognition in real-world videos [20, 19, 13]. In addition, this approach is easy to apply to new problems since it does not use any domain-specific features or prior domain knowledge.

First, a set of spatial temporal interest points (STIP) are extracted from a video clip. At each interest point, we extract a HoG (Histograms of oriented Gradients) feature and a HoF (Histograms of optical Flow) feature computed on the 3D video space-time volume. The patch is partitioned into a grid with 3x3x2 spatio-temporal

blocks. Four-bin HoG and five-bin HoF descriptors are then computed for all blocks and concatenated into 72-element and 90-element descriptors, respectively. We then concatenate these vectors to form a 162-element descriptor. A randomly sampled set of 500,000 motion descriptors from all video clips is then clustered using K-means(k=200) to form a vocabulary or "visual codebook." Finally, a video clip is represented as a histogram over this vocabulary. The final "bag of visual words" representing a video clip consists of a vector of k values, where the ith value represents the number of motion descriptors in the video that belong to the ith cluster. Figure 4 shows some sample frames with detected motion features. As shown, most motion features are detected on interesting and useful patches which form an integral part of the activity. A problem with this approach is that STIP points are extracted from the background when there is camera movement or a moving background. Many activities are difficult to distinguish, such as riding and driving. As discussed later, objects in the video can provide useful additional context for correctly identifying activities.

We then use the labeled clip descriptors to train an activity classifier. We tried several standard supervised classification methods from WEKA [34]. We achieved the best results using bagged REP decision trees. The trained classifier provides an initial probability distribution over activity labels for each test video based on spatio-temporal features.

Figure 4. Spatio-Temporal Interest Points

3.3 Object Detection in Videos

We used an off-the-shelf pre-trained object detector based on *Discriminatively Trained Deformable Part Models* [12] to detect objects in videos (release 4.01 [11]). We used models for 19 objects pre-trained on the "trainval" data for the PASCAL Visual Object Classes Challenge 2009 [9]. The 19 objects detected are: *aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, potted plant, sheep, sofa, train, tv monitor*. This approach provides robust, state-of-the-art object detection for static images, which we adapted to videos as follows.

First, we extracted one frame per second from each video in the test set, and represented each video by a set of frames. Then we ran the object detector on each resulting frame to produce bounding-boxes with scores for each of the 19 objects. Figure 5 shows some sample object detections. We used *Platt scaling* [29] to map these scores to calibrated probabilities. To produce a probability, $P(O_i|V_j)$, for object O_i appearing in video V_j, we took the maximum probability assigned to *any* detection of object O_i in *any* of the frames for video V_j. In this way, we computed a probability of each of the 19 objects occurring in each video. This approach was effective for our test corpus; however, exploring other techniques for using an object detector trained on static images to detect objects in video is another area for future research.

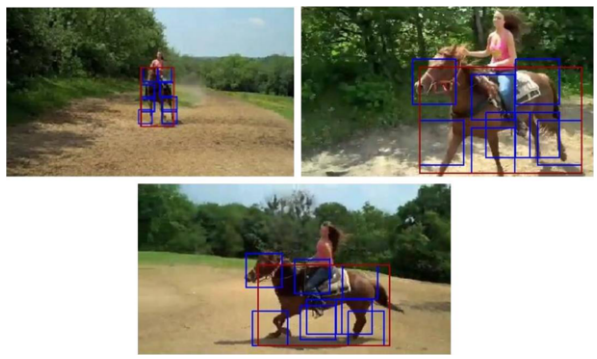

Figure 5. Horse Detections

3.4 Learning Correlations between Activities and Objects

To gather information on the correlation between activities and objects we mined the 2005 English Gigaword corpus, a comprehensive archive of newswire text assembled by the Linguistic Data Consortium (LDC) [5]. We used five distinct international sources of English newswire containing a total of 6 million documents or 15 GB of raw text. Using this corpus, we computed occurrence and co-occurrence counts for activities and objects. An occurrence of an activity A_i was defined as an occurrence (after stemming) of any one of the verbs in the verb cluster defining A_i. An occurrence of an object O_i was defined as any occurrence (after stemming) of the noun defining O_i or a "synonym." The set of "synonyms" was found by considering all nouns in the descriptive sentences in the training data and keeping those whose *Lesk similarity* [22] with the defining noun was greater than 0.5. This approach was found to be more effective than just using synonyms from WordNet.

In order to test the utility of different levels of language processing, we used one of four different methods for determining the co-occurrence of an activity and an object. In the first approach, called *windowing*, an activity and an object were said to co-occur if and only if there was an occurrence of the object within w or fewer words of an occurrence of the activity (after removing stop words). We tried window sizes, w, of 3, 10 and the entire sentence.

In the second method, called *POS tagging*, we first Part-of-Speech (POS) tag the English Gigaword corpus using the Stanford tagger [33]. Then, an activity and an object were said to co-occur if and only if there was an occurrence of the object tagged as a Noun within w or fewer words of an occurrence of the activity tagged as a Verb (after removing stop words). We tried window sizes, w, of 3, 10 and the entire sentence.

In the third approach, called *parsing I*, an activity and an object were said to co-occur if and only if the object was the direct object of the activity verb in the sentence. In the fourth approach, called *parsing II*, an activity and an object were said to co-occur if the object is syntactically attached to the activity verb by any relevant grammatical relation (e.g. ADVP, PP, NP). We used the Stanford Parser [2] to produce a typed dependency parse tree for each sentence. Given this

parse tree, it is easy to check if the object appears as the direct object of the activity verb, or is attached to the activity verb by any other relation. By comparing the parsing, POS tagging, and windowing approaches in the experiments below, we evaluated the advantage of performing deeper language processing to determine the correlation between activities and objects. For example, consider the following sentence from the English Gigaword corpus: "Sitting in Bethlehem, PA., cafe, Kaye thumps a table and wails white blues." Using the windowing or POS-tagging approach (with one of the larger windows), we would determine that "sit" and "table" co-occur. However, if we apply one of the parsing methods, we would find that "table" is neither a direct object of sit, nor is it attached by any other grammatical relation, and thus they would not be considered to co-occur.

Using counts of these occurrences and co-occurrences in Gigaword, we estimated the probability of each activity given each object using Laplace (add-one) smoothing as follows:

$$P(A_i|O_j) = (Count(A_i, O_j) + 1)/(Count(O_j) + |A|) \quad (1)$$

where $Count(A_i, O_j)$ is the number of co-occurrences of A_i and O_j, $Count(O_j)$ is the number of occurrences of O_j, and $|A|$ is the total number of activities.

3.5 Integrated Activity Recognizer

Using the information for detected objects along with the correlation between activities and objects, we obtain an "object-based" probability distribution over activity labels for each test video. Our final activity recognizer combines this distribution with the initial probability distribution over activity labels based on spatio-temporal features.

Let the features used by the object detector be denoted as F_o. We compute the probability of an activity A_i given these object features by applying chain rule as follows:

$$P(A_i|F_o) = \sum_{j=1}^{|O|} P(A_i|O_j) * P(O_j|F_o) \quad (2)$$

where $|O|$ is the total number of object detectors (19 for our system). The first component $P(A_i|O_j)$ is provided by the text mining component described in Section 3.4, and the second component $P(O_j|F_o)$ is provided by the object detector described in Section 3.3.

As discussed in Section 3.2, a classifier trained on the spatio-temporal features of the video (denoted as F_v) gives us an initial probability distribution over activities labels in each test video, $P(A_i|F_v)$. To recognize the activity in a test video, we combine both distributions $P(A_i|F_o)$ and $P(A_i|F_v)$ as follows. The final recognized activity is:

Videos on which object detector detected at least one object

$$= \operatorname{argmax}_i P(A_i|F_o, F_v) \quad (3)$$

$$= \operatorname{argmax}_i P(F_o, F_v|A_i) * P(A_i)/P(F_o, F_v) \quad (4)$$

by Bayes' Rule

$$= \operatorname{argmax}_i P(F_o, F_v|A_i) * P(A_i) \quad (5)$$

by removing terms that are same for all i

$$= \operatorname{argmax}_i P(F_o|A_i) * P(F_v|A_i) * P(A_i) \quad (6)$$

by assuming features are independent given the activity

$$= \operatorname{argmax}_i P(A_i|F_o) * P(F_o) * P(A_i|F_v) * P(F_v)/P(A_i) \quad (7)$$

by Bayes' Rule + algebra

$$= \operatorname{argmax}_i P(A_i|F_o) * P(A_i|F_v)/P(A_i) \quad (8)$$

by removing terms that are same for all i

Videos on which there were no detected objects

$$= \operatorname{argmax}_i P(A_i|F_v) \quad (9)$$

Thus we consider only $P(A_i|F_v)$ when no object is detected and $P(A_i|F_o, F_v)$ when objects are recognized. Note that this derivation assumes that the video and object features are independent given the activity class. This is an instance of the "naive Bayes" assumption, which, although rarely completely justified, has been demonstrated to be very useful in practice [7]. In this way, we combine information from STIP features of the video with object-detection features in the static images. This integration could possibly also improve object detection in video; however, in this paper we focus on activity recognition, a difficult task which is less well-studied than object recognition, and therefore more likely to benefit from the integration of the two.

4 Experimental Evaluation

This section presents an experimental evaluation of our approach. First we describe the dataset, next we explain our experimental methods, and finally we present the results.

4.1 Dataset

We used the data collected by Chen et al. [3], consisting of 1,970 short real-world YouTube video clips accompanied with about 122K natural-language descriptions. The videos were collected by naive workers recruited using Amazon Mechanical Turk (AMT). They were given instructions to submit YouTube clips that were short, have a single unambiguous event, and could be described by a single natural-language sentence. Natural-language descriptions in more than 16 languages for these videos were then collected from additional AMT workers. We used only the English descriptions which total about 85k. In the future, we would like to extract more information from the descriptions in other languages. Each video clip is approximately 10 seconds long. Much of the work on activity recognition is performed on simple "staged" videos with a single person performing a very scripted activity. By comparison, this YouTube data has more complexity, diversity, and noise, making for a difficult real-world activity-recognition corpus.

4.2 Experimental Method

We divided this data set into disjoint training and test sets. In order to provide a useful test of the effect of object recognition on activity detection given the limitations of our off-the-shelf object recognizer, we selected test videos that were likely to contain one of the 19 object classes covered by the object recognizer. We first found all videos that included a reference to at least one of these 19 objects in their English descriptions. A description was determined to reference an object if included the defining word for the object or one of its synonyms as described in Section 3.4. The rest of the data was then used to discover activity classes by clustering the original 265 verb stems used to describe these videos (see Section 3.1). After manually cutting the resulting hierarchical clustering at a level to create meaningful activity classes, and removing classes with fewer

than 9 videos, this left a labeled training set containing 28 activity classes with a total of 1,119 videos.

Finally, a disjoint test set of 128 videos was assembled by selecting videos that both contained a reference to one of the 19 object classes and belonged to one of these 28 activity classes (as determined by the verbs used to describe these test videos). It is important to note that the system does not use *any* of the information from the linguistic descriptions of a test video when predicting the activity it depicts; classification is based solely on the video. The linguistic descriptions of the test videos are only used to *evaluate* the accuracy of activity recognition. Also, training does not use any information from *either* the test videos or their linguistic descriptions.

The training data is used to construct an initial activity classifier based on spatio-temporal features as described in Section 3.2. The resulting model is used to predict an initial activity distribution for each of the test videos. Next, we perform object detection on the 128 test videos as described in section 3.3. This gives us probability of each of the 19 objects appearing in each test video. For obtaining correlations between activities and objects, we evaluated the *windowing*, *POS tagging*, *parsing I* and *parsing II* methods as described in Section 3.4 and compared their results. Finally, all this information is combined to make a final activity prediction for the test videos as described in Section 3.5. Since the test videos were specifically chosen to refer to the target objects, the distribution of activity classes in the training and test data are different. Therefore, we found we got better results using a simple uniform prior over activities ($P(A_i)$) in Equation 8 rather than one estimated from the training data. This effectively removes this term from the equation since it is the same for all classes.

We evaluated predictions on the test set by measuring classification Accuracy and Weighted Average Precision (WAP). The precision for a given activity class is the fraction of the videos assigned to the class which are correctly classified. WAP is the average precision across all classes, weighted by the number of test instances in each class.

4.3 Results

4.3.1 Results using only Spatio-Temporal Interest Points

Table 1 shows results of activity recognition using just the STIP features of the video [18] described in 3.2. Results are shown for various Weka classifiers. Bagged REP decision trees gave the best results, so we used that approach in the final system.

Table 1. Recognition Results using only Spatio-Temporal Interest Points

Classifier	Accuracy	WAP
Decorate	0.289	0.361
Bagged REP Tree	**0.3906**	**0.392**
Bagged J48 Decision Tree	0.375	0.387
AdaBoost	0.25	0.314

4.3.2 Results for Object Detection in Videos

To evaluate the accuracy of the object detector on our data, we manually determined the number of correct and incorrect detections of each object in the test data. For each object class, Table 2 shows the numbers of correct and incorrect detections as well as the number of videos actually containing the object. Objects for which there are no correct or incorrect detections are not shown in the table. Note that the aeroplane recognizer detects many false positives. The system also confused similar objects, such as motorbike and bicycle, car and bus, etc., which is understandable. But wrongly detecting a bicycle as a motorbike or vice versa, does not usually hurt the final activity results, because the same activities tend to be correlated with such easily confusable objects. For example, the most common verb correlated with both motorbike and bicycle is "ride."

Table 2. Results of Object Detection in Videos

Object Model	True Positives	False Positives	Videos with Object
horse	24	0	33
car	15	12	22
motorbike	7	3	25
bicycle	4	6	8
aeroplane	2	28	3
tvmonitor	2	2	3
bottle	1	0	4
bus	0	2	1
train	0	2	2

4.3.3 Results for Our Integrated Activity Recognizer

Final results for our full system using different text-mining approaches are shown in Table 3. Compared to the initial results in Table 1, all of the approaches demonstrate the advantage of using object recognition and text-mined activity/object associations to improve activity recognition. The best results are produced when parsing is used to identify direct objects, demonstrating the advantage of deeper language processing. Results improve for activities like *ride*, *fly* and *drive* which have direct objects like *horse*, and *bicycle* in case of *ride*, *plane* in case of *fly* , *car* in case of *drive*, etc. Using the *parsing I* approach, we get 67 of 128 test videos correct as opposed to 50 correct when just using STIP features. Since the aeroplane detector gives many false positives, when we deleted this detector, the results improved to 69 correctly-classified videos.

Table 3. Final Results Using Different Text-Mining Methods

Correlation Method	Window	Accuracy	WAP
windowing	3	0.4687	0.4568
windowing	10	0.4687	0.4568
windowing	full sentence	0.4609	0.4551
POS tagging	3	0.4609	0.4617
POS tagging	10	0.4375	0.4565
POS tagging	full sentence	0.3984	0.4542
parsing I	full sentence	**0.5234**	**0.4987**
parsing II	full sentence	0.4844	0.3823

Table 4 presents results comparing the integrated system to its individual components, just using video STIP features and just using object information. The results demonstrate that integrating both sources of information significantly improves results compared to using either one alone.

Table 4. Results for System Ablations

Method	Accuracy
Only $P(A_i\|F_v)$	0.3906
Only $P(A_i\|F_o)$ *with parsing I*	0.3828
Integrated System	**0.5234**

5 Future Work

The current approach could be extended and improved in many ways. With regard to activity discovery, the use of Word Sense Disambiguation could improve the similarity measure used to cluster verbs. Also, the selection of the appropriate number of clusters needs to be automated. Our approach could also be tested on other activity recognition datasets such as the movie-script data used in [19].

With respect to object recognition, models for detecting a much broader set of object classes should be utilized. Additional object detectors could be trained using the bounding boxes from ImageNet [6]. A larger set of object detectors would allow extending our experiments to a larger subset of test videos from our YouTube corpus. Also, object-detection methods specifically designed for video which use optical flow to help detect object boundaries could be useful. Finally, it would be useful to explore how an integrated approach like ours could improve object detection in videos by using information from activity recognition.

Alternative approaches to integrating spatio-temporal activity recognition, object detection, and text-mined activity-object correlations should also be explored. Our simple "naive Bayesian" integration works fairly well, but approaches that model the dependencies between object and video features could potentially work better.

Finally, our ultimate goal is to construct a system that can produce complete natural-language sentences for describing videos. This will require detecting all the arguments of an activity such as subjects, direct objects, and objects of prepositions; as well as full natural-language sentence generation.

6 Conclusions

This paper has made three important contributions to video activity recognition. First, it has introduced a novel method for automatically discovering activity classes from natural-language descriptions of videos. Second, it has demonstrated how existing activity-recognition systems can be improved using object context together with correlations between objects and activities. Third, it has shown how language processing can be used to automatically extract the requisite knowledge about the correlation between objects and activities from a corpus of general text. Finally, we integrated these components to produce an activity recognizer that improves accuracy on a realistic video corpus by more than 10 percentage points over a standard activity recognizer using the features described in [18].

ACKNOWLEDGEMENTS

This work was funded by National Science Foundation grants IIS–0712097 and IIS–1016312. We would like to thank Kristen Grauman and the anonymous reviewers for their valuable comments.

REFERENCES

[1] J. K. Aggarwal and S. Park, 'Human motion: Modeling and recognition of actions and interactions', in *3DPVT*, (2004).
[2] Marie catherine De Marneffe, Bill Maccartney, and Christopher D. Manning, 'Generating typed dependency parses from phrase structure parses', in *LREC*, (2006).
[3] David L. Chen and William B. Dolan, 'Collecting highly parallel data for paraphrase evaluation', in *ACL*, (2011).
[4] Timothée Cour, Chris Jordan, Eleni Miltsakaki, and Ben Taskar, 'Movie/script: Alignment and parsing of video and text transcription', in *ECCV*, (2008).
[5] Ke Chen and Kazuaki Maeda David Graff, Junbo Kong, 'English gigaword second edition', in *LDC*, (2005).
[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li, 'Imagenet: A large-scale hierarchical image database', in *CVPR*, (2009).
[7] Pedro Domingos and Michael Pazzani, 'On the optimality of the simple Bayesian classifier under zero-one loss', *ML*, (1997).
[8] Alexei A. Efros, Alexander C. Berg, Er C. Berg, Greg Mori, and Jitendra Malik, 'Recognizing action at a distance', in *ICCV*, (2003).
[9] Mark Everingham, Luc J. Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman, 'The pascal visual object classes (voc) challenge', *IJCV*, (2010).
[10] Mark Everingham, Josef Sivic, and Andrew Zisserman, 'Hello! my name is... buffy" – automatic naming of characters in tv video', in *BMVC*, (2006).
[11] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. http://people.cs.uchicago.edu/ pff/latent-release4/.
[12] Pedro F. Felzenszwalb, Ross B. Girshick, David A. McAllester, and Deva Ramanan, 'Object detection with discriminatively trained part-based models', *IEEE Trans. Pattern Anal. Mach. Intell.*, (2010).
[13] Adrien Gaidon, Marcin Marszalek, and Cordelia Schmid, 'Mining visual actions from movies', in *BMVC*, (2009).
[14] Abhinav Gupta and Larry S. Davis, 'Objects in action: An approach for combining action understanding and object perception', in *CVPR*, (2007).
[15] Sonal Gupta and Raymond J. Mooney, 'Using closed captions as supervision for video activity recognition', in *AAAI*, (2010).
[16] Anthony Hoogs and A. G. Amitha Perera, 'Video activity recognition in the real world.', in *AAAI*, (2008).
[17] Jay J. Jiang and David W. Conrath, 'Semantic similarity based on corpus statistics and lexical taxonomy', *CoRR*, (1997).
[18] Ivan Laptev, 'On space-time interest points', *IJCV*, (2005).
[19] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld, 'Learning realistic human actions from movies', in *CVPR*, (2008).
[20] Ivan Laptev and Patrick Pérez, 'Retrieving actions in movies', in *ICCV*, (2007).
[21] C. Leacock and M. Chodorow, 'Combining local context and WordNet similarity for word sense identification', in *WordNet: An Electronic Lexical Database*, (1998).
[22] Michael Lesk, 'Automatic sense disambiguation: How to tell a pine cone from an ice cream cone', in *SIGDOC*, (1986).
[23] Dekang Lin, 'An information-theoretic definition of similarity', in *ICML*, (1998).
[24] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, 2008.
[25] Christopher D. Manning and Hinrich Schütze, *Foundations of Statistical Natural Language Processing*, 1999.
[26] Marcin Marszalek, Ivan Laptev, and Cordelia Schmid, 'Actions in context', in *CVPR*, (2009).
[27] Darnell J. Moore, Irfan A. Essa, and Monson H. Hayes, 'Exploiting human actions and object context for recognition tasks', in *ICCV*, (1999).
[28] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi, 'Wordnet: : Similarity - measuring the relatedness of concepts', in *AAAI*, (2004).
[29] John C. Platt, 'Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods', in *Advances in Large Margin Classifiers*, (1999).
[30] Philip Resnik, 'Using information content to evaluate semantic similarity in a taxonomy', in *IJCAI*, (1995).
[31] M. S. Ryoo and J. K. Aggarwal, 'Hierarchical recognition of human activities interacting with objects', in *CVPR*, (2007).
[32] Christian Schüldt, Ivan Laptev, and Barbara Caputo, 'Recognizing human actions: A local svm approach', in *ICPR*, (2004).
[33] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer, 'Feature-rich part-of-speech tagging with a cyclic dependency network', in *HLT-NAACL*, (2003).
[34] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations (2nd edition)*, 2005.
[35] Jianxin Wu, Adebola Osuntogun, Tanzeem Choudhury, Matthai Philipose, and James M. Rehg, 'A scalable approach to activity recognition based on object use', in *ICCV*, (2007).
[36] Zhibiao Wu and Martha Stone Palmer, 'Verb semantics and lexical selection', in *ACL*, (1994).