

Supplementary material: Spatio-temporal Person Retrieval via Natural Language Queries

Masataka Yamaguchi¹, Kuniaki Saito¹, Yoshitaka Ushiku¹, Tatsuya Harada^{1,2}

¹Graduate School of Information Science and Technology, The University of Tokyo ²RIKEN
{yamaguchi, ksaito, ushiku, harada}@mi.t.u-tokyo.ac.jp

A. Dataset Statistics

In this section, we provide the further details of the dataset statistics.

The description length.

We first analyze the description length (i.e., the number of words in a description). Figure 1 shows the distribution of the number of words in a description. We can see that our dataset contains various lengths of descriptions. The average length of descriptions in our dataset is 13.1. We also show the comparison of the average description length of our dataset to those of other datasets in Table 1. ReferIt [7] and Google RefExp [12] are the datasets of referring expressions, each of which is true of only a single region in an image. The descriptions in VisualGenome [10], MSR-VTT [20] and MSCOCO [2] focus on regions in images, whole images and videos, respectively. Even though a description in our dataset focuses on a single person, the average description length of our dataset is larger than not only those of the datasets of which descriptions focus on regions in images, but also those of the datasets of which descriptions focus on the whole images or videos. This implies that the descriptions in our dataset tend to contain more detailed information than those in other datasets.

The number of annotated people in a clip.

Figure 2 shows the distribution of the number of people who are annotated with bounding boxes and descriptions in a single clip. While many clips contain only one annotated person, some clips contain multiple annotated people.

The number of occurrences of each high-frequency word.

Figure 4 shows the number of occurrences of the most frequently occurring words (Stop words are excluded). We can see that high-frequency words involve various types of words such as colors, actions, clothes and places.

Figure 5 shows the comparison of frequencies of words in Figure 4 between our dataset and VisualGenome. While the frequencies of words describing colors (e.g. *black*, *white*, *blue* and *red*) and people (e.g. *man*, *woman*, *girl* and *boy*) in our dataset are close to those in VisualGenome, the

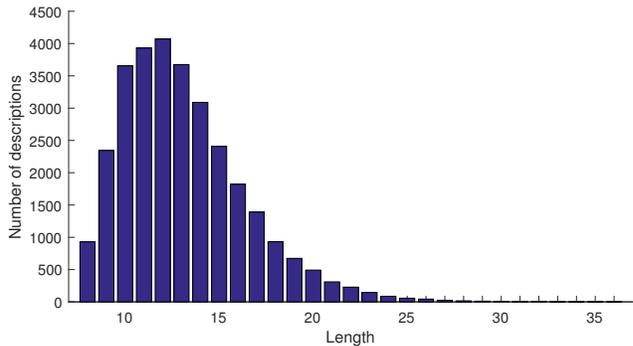


Figure 1. A distribution of the number of words in a description.

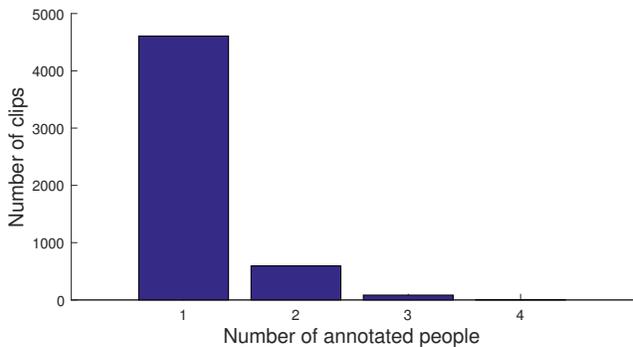


Figure 2. A distribution of the number of people who are annotated with bounding boxes in a clip.

frequencies of words describing some actions (e.g. *playing*, *performs*, *performing*, *dancing*, *using*, *jump* and *wrestling*) in our dataset are much higher than those in VisualGenome.

Comparison to the dataset in [11].

Table 2 shows the comparison of our dataset to the dataset used in [11]. In terms of the number of clips, objects / people, and descriptions, and the average length of descriptions, our dataset outperforms the one used in [11].

Dataset	Type of descriptions	The average length
ReferIt [7]	Referring expressions	3.6
LSMDC 16 [16]	Video descriptions	4.1
VisualGenome [10]	Descriptions of regions in images	5
Google RefExp [12]	Referring expressions	8.4
MSR-VTT [20]	Video descriptions	9.3
MSCOCO (train & val) [2]	Image descriptions	10.5
Ours	Descriptions of people in videos	13.1

Table 1. Comparison of the average description lengths among datasets.

	#Clip	Average clip length	#Object/Person	#Description	Average description length	Data source
[11]	21	381 frames	1,068	443	7.9	KITTI [4]
Ours	5,293	260 frames	6,073	30,365	13.1	ActivityNet [1]

Table 2. Comparison of our dataset to the dataset used in [11]. Since some descriptions in [11] describe multiple objects, the number of objects annotated with descriptions is larger than that of descriptions.

B. Settings of retrieval methods

In this section, we describe the settings of DSPE and DSPE++ used in Section 5, Section C.1 and Section C.2.

For both DSPE and DSPE++, we set $\alpha_1 = 2$, $\alpha_2 = 0$, $\alpha_3 = 0.2$ and $m = 0.9$. For DSPE++, we set $\alpha_4 = 50$. For training models, we use SGD with mini-batches of 1500 pairs, momentum of 0.9 and weight decay of 0.0005. We train models for 600 iterations. We use a learning rate 0.01 for the first 500 iterations, and we use a learning rate 0.001 for the remaining 100 iterations. For early stopping, we test models on the validation dataset after every epoch, and choose the best one for computing the accuracy on the test dataset.

We implemented DSPE and DSPE++ using Chainer [18], in which GRU is implemented.

C. Additional Experiments

C.1. Comparison of Text Features

In this section, we discuss the experiments to compare three types of text features. In these experiments, we use mean pooling and DSPE++ for pooling segment-level features and retrieval, respectively. We compare the three following types of text features:

FV based on HGLMM. The first one is FV based on HGLMM [9]. To compute the FVs based on HGLMM, we first apply Independent Component Analysis (ICA) for 300-dimensional word2vec vectors¹ [14] and train an HGLMM with 30 centers using ICA-applied word vectors. Next, we compute the FVs of descriptions using the learned HGLMM and apply power and L2 normalizations to them. We also apply PCA to them, as we obtained higher retrieval accuracy than when using the original FVs. We set the number

¹<https://code.google.com/archive/p/word2vec/>

of dimensions after reduction to 1,000 based on the results obtained in the validation dataset.

Skip-thought vectors. The second one is skip-thought vectors [8]. To extract skip-thought vectors, we use the model provided by the authors². As with the case using FV, we reduce the features to 1000 dimensions by applying PCA to them.

Features encoded by GRU. As the third choice, we extract text features h by encoding a given sentence using GRU trained from scratch. Specifically, given a sequence of N words in one-hot form w_1, w_2, \dots, w_N , we extract its text features $h = h_N$ by iterating the following operation from $k = 1$ to $k = N$:

$$x_k^{emb} = W_{emb}w_k, \quad (1)$$

$$r = \sigma(W_r x_k^{emb} + U_r h_{k-1}), \quad (2)$$

$$z = \sigma(W_z x_k^{emb} + U_z h_{k-1}), \quad (3)$$

$$\bar{h} = \tanh(W_h x_k^{emb} + U_h (h_{k-1} \odot r)), \quad (4)$$

$$h_k = (1 - z) \odot h_{k-1} + z \odot \bar{h}, \quad (5)$$

where $h_0 = 0$. In the following, we call this feature extraction method as GRU encoding. When using GRU encoding, we train a GRU module for feature extraction and embeddings for retrieval simultaneously on the training dataset. We set the dimension of a word vector x_k^{emb} and a hidden state h_k as 300 and 256, respectively.

In principle, GRU encoding is close to the feature extraction process of skip-thought vectors. However, these are different in that a GRU module for GRU encoding is trained from scratch on our dataset in a supervised manner, while GRU modules in skip-thought vectors are trained on another large-scale dataset in an unsupervised manner.

²<https://github.com/ryankiros/skip-thoughts>

Text Features	R@1	R@5	R@10
FV	0.357	0.702	0.795
Skipthought Vector	0.276	0.623	0.744
GRU encoding	0.286	0.610	0.741
FV + Skipthought Vector	0.352	0.682	0.781
FV + GRU encoding	0.352	0.669	0.776

Table 3. Performance comparison of text features.

The results are shown in Table 3. In all metrics, using FV achieves higher accuracy than using skip-thought vectors or features obtained by GRU encoding. We also conduct experiments using features obtained by concatenating FV with skip-thought vectors (FV + Skipthought Vectors), or concatenating FV with features obtained by GRU encoding (FV + GRU encoding), but using FV alone achieves the highest accuracy.

C.2. Comparison of Feature Aggregation

In this section, we discuss the experiments to compare three feature aggregation methods. In these experiments, we use FV based on HGLMM as text features, and use DSPE++ for retrieval. In the following, a sequence of segment-level features is denoted by x_1, x_2, \dots, x_N , and the aggregated feature vector is denoted by h . We compare the three following aggregation methods:

Mean Pooling. The first one is mean pooling, which encodes features as follows:

$$h = \frac{1}{N} \sum_i^N x_i. \quad (6)$$

Max Pooling. The second one is max pooling, which encodes features as follows:

$$h = \max(x_1, x_2, \dots, x_N). \quad (7)$$

Aggregation with GRU. As the third choice, we encode a sequence of features using GRU. More specifically, we obtain the aggregated features $h = h_N$ by iterating the following operation from $k = 1$ to $k = N$:

$$r = \sigma(W_r x_k + U_r h_{k-1}), \quad (8)$$

$$z = \sigma(W_z x_k + U_z h_{k-1}), \quad (9)$$

$$\bar{h} = \tanh(W_h x_k + U_h (h_{k-1} \odot r)), \quad (10)$$

$$h_k = (1 - z) \odot h_{k-1} + z \odot \bar{h}, \quad (11)$$

where $h_0 = 0$. As with Section C.1, we also refer to this strategy as GRU encoding. When using GRU encoding, we train a GRU module for encoding and embeddings for retrieval simultaneously on the training dataset. We set the dimension of a hidden state h_k as 512.

Feature Aggregation Strategy	R@1	R@5	R@10
Mean	0.357	0.702	0.795
Max	0.326	0.667	0.762
GRU encoding	0.277	0.584	0.684
Mean + Max	0.355	0.702	0.798
Mean + GRU encoding	0.357	0.692	0.791

Table 4. Performance comparison of feature encoding methods.

α_4	R@1	R@5	R@10
0	0.347	0.687	0.783
5.0	0.359	0.684	0.779
50.0	0.357	0.702	0.795
500.0	0.076	0.254	0.380

Table 5. Comparison of α_4 in DSPE++.

The results of experiments are shown in Table 4. In all metrics, using mean pooling achieves the highest accuracy among three aggregation methods. We also conduct experiments using the combinations of mean pooling and max pooling, or mean pooling and GRU encoding, but no strategy outperforms mean pooling in all of three metrics. This result suggests that the information of the maximum, or the order of features used in this experiment is relatively less discriminative compared to the mean of features in this task.

C.3. Comparison of α_4 in DSPE++

We show the comparison of α_4 in DSPE++ in Table 5.

C.4. DSPE++ on Image-Sentence Retrieval

In this section, we discuss the experiments to compare DSPE and DSPE++ on image-sentence retrieval.

Settings. We conduct experiments on Flickr8k [6] and Flickr30k [21]. Flickr8k and Flickr30k consist of 8,000 and 31,783 images, respectively. In both datasets, five descriptions are annotated to each image. For Flickr8k, we use the standard dataset partition. For Flickr30k, we use the dataset partition used by [13]³. The numbers of training and testing images in Flickr8k are 6,000 and 1,000, respectively. The numbers of training and testing images in Flickr30k are 29,783 and 1,000, respectively.

To extract image and text features, we follow the process in [19]. To extract image features, we use the outputs from the fc7 layer in the VGG-16 layer net [17] pretrained on ImageNet [3]. We crop a given image in 10 ways and use the means of the features extracted from the 10 cropped regions as features for the image. The process of text feature extraction is explained in Section 4.3.2. We set the number of dimensions after dimension reduction to 6,000.

To embed features of both modalities into a common

³http://www.stat.ucla.edu/~junhua.mao/attachments/flickr30K_train_val_test_img_list.zip

space, we use the network architecture shown in Table 2 in the main paper. For both DSPE and DSPE++, we set $\alpha_1 = 2$, $\alpha_2 = 0$, $\alpha_3 = 0.2$. We set the margin m as 0.1 or 0.9. For DSPE++, we set $\alpha_4 = 10$.

To train models, we follow the training process in [19]. We use SGD with mini-batches of 1500 pairs, momentum of 0.9 and weight decay of 0.0005. We start training with a learning rate of 0.1, and decay it by 0.1 after every ten epochs. For testing, we use models obtained right after 30 epochs.

For these experiments, we use the code provided by the authors of [19]⁴.

Results. The results are shown in Table 6. In Flickr8k, DSPE++ ($m = 0.9$) outperforms the others in five out of six metrics. This result suggests the effectiveness of using the proposed loss in Flickr8k. In contrast, DSPE ($m = 0.1$) outperforms the others in all metrics in Flickr30k. From these results, it can be concluded that DSPE++ works better than DSPE in most datasets including our dataset and Flickr8k, while it does not do so in other datasets including Flickr30k. Considering that the loss of DSPE is the special case of that of DSPE++ (namely, the case setting $\alpha_4 = 0$ in the loss of DSPE++), to obtain high retrieval accuracy, DSPE++ rather than DSPE should be used while tuning the hyperparameter α_4 of DSPE++ on the validation dataset.

D. Recall Rate of Ground-truth Tubes

We show the recall-rate curve of ground-truth tubes in the validation dataset in Figure 3.

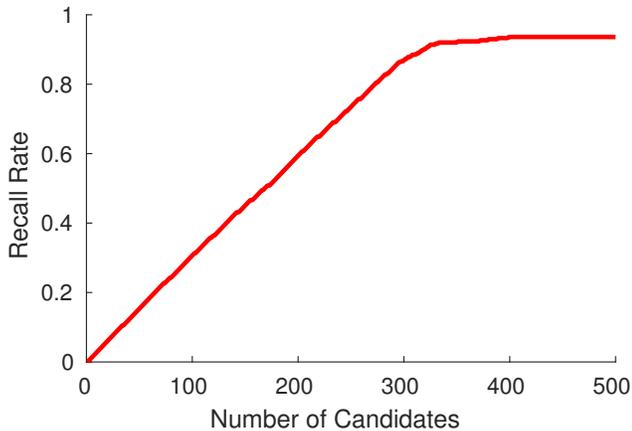


Figure 3. Recall rate of ground-truth tubes on the validation dataset.

⁴<https://dl.dropboxusercontent.com/u/17926179/embedding/embedding.htm>

E. Retrieved Examples

Examples of correctly retrieved people are shown in Figure 6. Further examples that are randomly chosen are shown in Figure 7 and Figure 8.

F. Action Detection on UCF Sports

In this section, we discuss the quantitative experiment of spatio-temporal action detection using our model. As explained in Section 5.3.2, we conduct the experiment on the UCF Sports dataset [15], which consists of 150 video clips with 10 action classes.

F.1. Settings

In this experiment, we detect tubes suitable for each action by just inputting the action category name to our model, which is trained on our dataset. We set the number of candidate tubes $N_c = 200$. We use FV based on HGLMM as the text features, and DSPE++ for retrieval. We use Average Precision (AP) as the evaluation metric. When computing AP, we assume dt_{tube} is the true positive of gt_{tube} in the case that the localization score $S_{loc}(gt_{tube}, dt_{tube})$ explained in Section 5.1 is over 0.5.

F.2. Results

Table 7 shows the result. Even though our model is trained without data in the UCF Sports dataset, the results of some classes are rather encouraging (e.g., 0.840 for “Lifting” and 0.435 for “Riding Horse”). These results suggest the versatility of our model.

However, the average precisions of some action classes are relatively low (e.g., 0.031 for “Swing Side”). This is considered to be due to the low specificity of each category name for its actual examples in the UCF Sports dataset.

In Table 7, we also show the detection results in [5]. Note that we cannot fairly compare the results in UCF Sports with other works including [5] since other works usually split the dataset into train/test and train models using the training part, while in our setting we use all videos as the test dataset and use no videos in this dataset for training.

References

- [1] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 2
- [2] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 1, 2
- [3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 3

Dataset	Method	margin	Image-to-Sentence			Sentence-to-Image		
			R@1	R@5	R@10	R@1	R@5	R@10
Flickr8k	DSPE (original)	-	-	-	-	-	-	-
	DSPE	0.1	0.258	0.512	0.642	0.185	0.444	0.598
	DSPE++	0.1	0.244	0.515	0.649	0.181	0.448	0.593
	DSPE	0.9	0.255	0.511	0.642	0.184	0.449	0.603
	DSPE++	0.9	0.241	0.516	0.652	0.188	0.455	0.608
Flickr30k	DSPE (original)	0.1	(0.403)	(0.689)	(0.799)	(0.297)	(0.601)	(0.721)
	DSPE	0.1	0.393	0.681	0.797	0.29	0.600	0.721
	DSPE++	0.1	0.372	0.660	0.786	0.284	0.595	0.711
	DSPE	0.9	0.370	0.663	0.765	0.263	0.562	0.687
	DSPE++	0.9	0.377	0.656	0.762	0.270	0.575	0.703

Table 6. Performance Comparison on Flickr8k and Flickr30k.

Action Category	AP (ours)	AP ([5])
Lifting	0.840	(1.00)
Riding Horse	0.435	(1.00)
Riding Skateboard	0.318	(0.417)
Diving	0.291	(1.00)
Running	0.223	(0.117)
Kicking	0.180	(0.667)
Walking	0.125	(0.458)
Swing Bench	0.106	(1.00)
Golf Swing	0.095	(0.917)
Swing Side	0.031	(1.00)

Table 7. The result of the experiment of action detection on the UCF Sports dataset using our model trained on our dataset.

- [4] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 2
- [5] G. Gkioxari and J. Malik. Finding action tubes. In *CVPR*, 2015. 4, 5
- [6] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *JAIR*, 47:853–899, 2013. 3
- [7] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 1, 2
- [8] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. Skip-thought vectors. In *NIPS*, 2015. 2
- [9] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *CVPR*, 2015. 2
- [10] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016. 1, 2, 6
- [11] D. Lin, S. Fidler, C. Kong, and R. Urtasun. Visual semantic search: Retrieving videos via complex textual queries. In *CVPR*, 2014. 1, 2
- [12] J. Mao, J. Huang, A. Toshev, O. Camburu, A. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. 1, 2
- [13] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). In *ICLR*, 2015. 3
- [14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013. 2
- [15] M. D. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008. 4
- [16] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, and B. Schiele. Movie description. *International Journal of Computer Vision*, 123(1):94–120, 2017. 2
- [17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 3
- [18] S. Tokui, K. Oono, and S. Hido. Chainer: a next-generation open source framework for deep learning. In *NIPS Workshop on Machine Learning Systems*, 2015. 2
- [19] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, 2016. 3, 4
- [20] J. Xu, T. Mei, T. Yao, and Y. Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 1, 2
- [21] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2014. 3

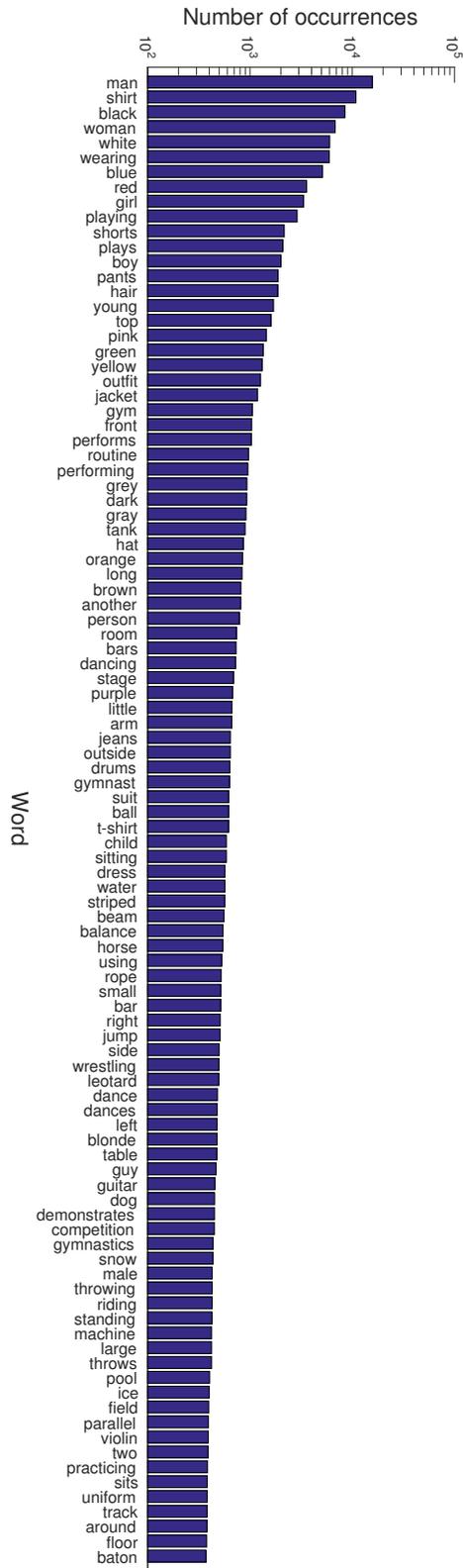


Figure 4. The numbers of occurrences of the top-100 most frequently occurring words. We exclude stop words.

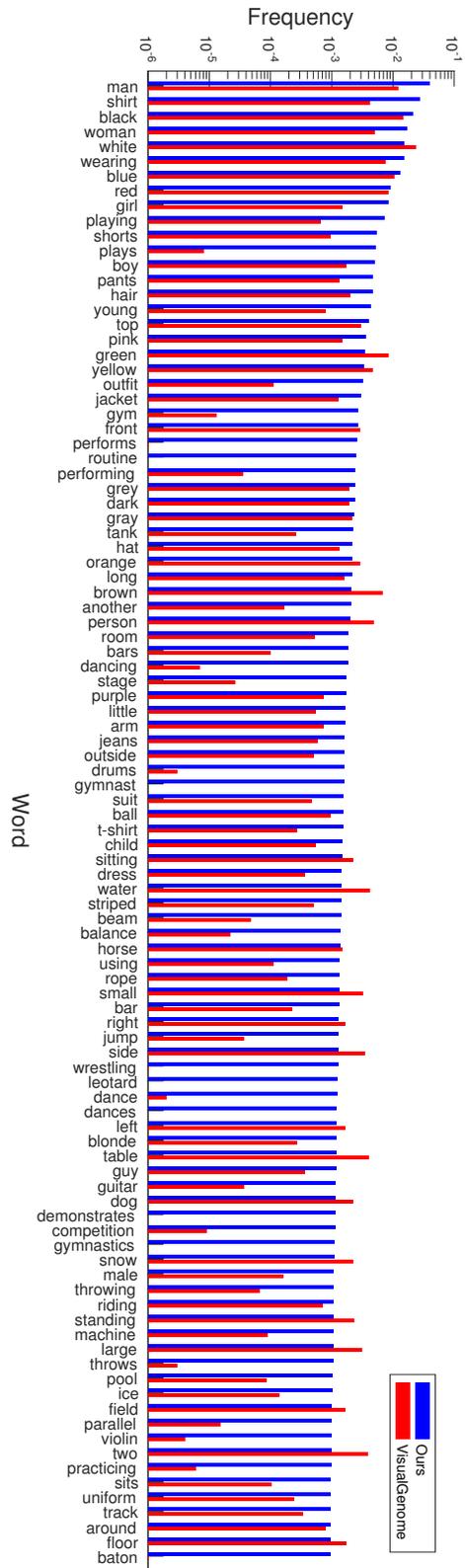


Figure 5. Comparison of the frequencies of the words in Figure 4 between our dataset and VisualGenome [10]. We exclude stop words. This figure is best viewed in color.

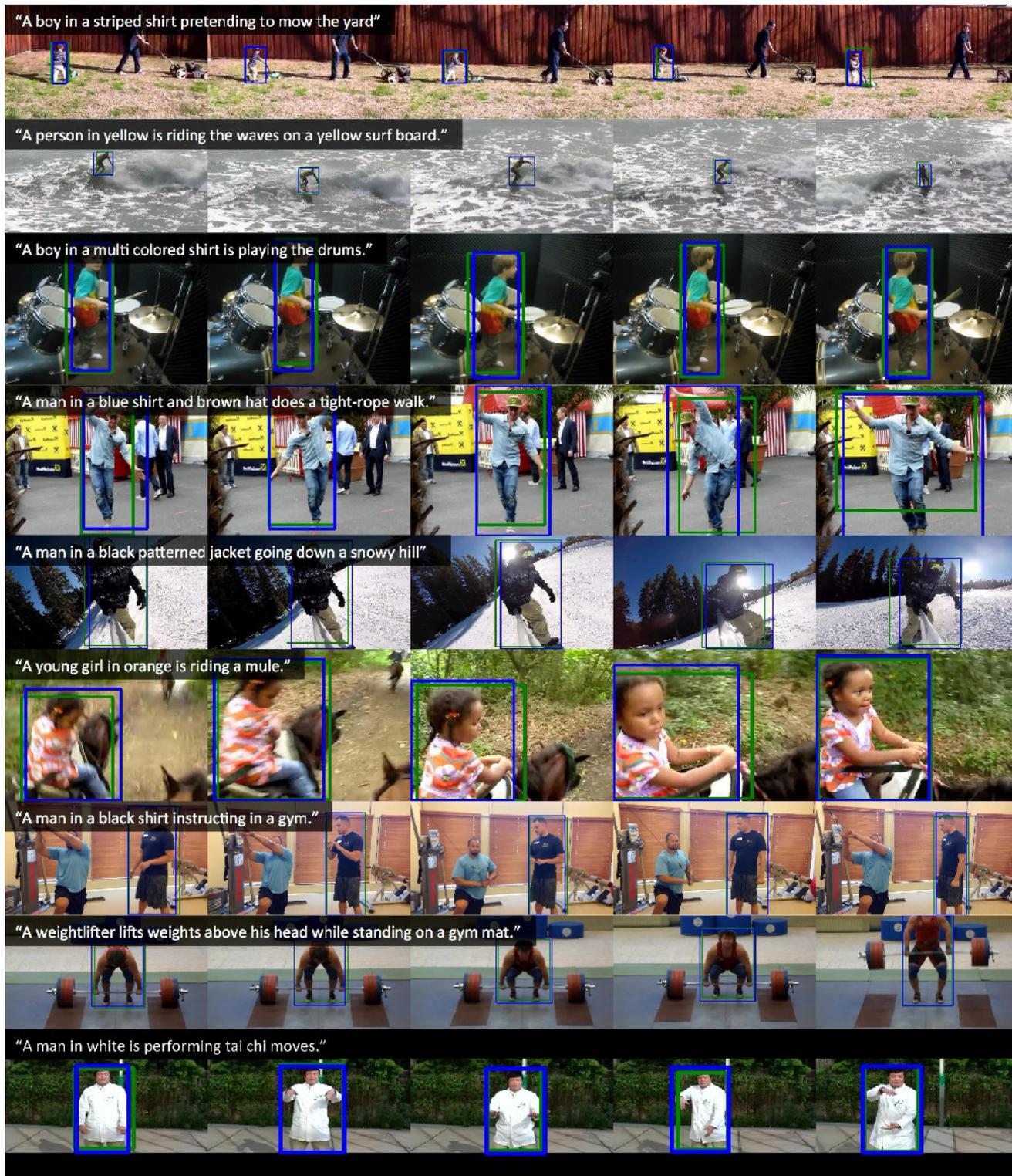
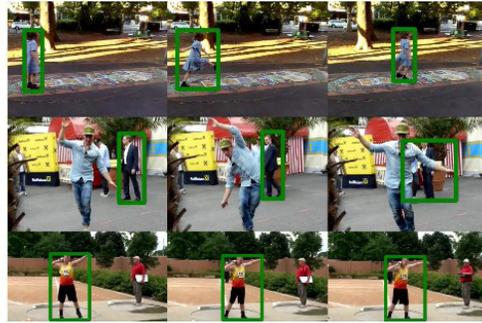


Figure 6. Examples of correctly retrieved people. The blue and green bounding boxes are the ground truth and the top-1 retrieved results, respectively. Note that the search space consists of 283 video clips and its total duration is 45.5 minutes. We show five frames for each example.

A person with long black hair hitting on a guitar in the streets



A boy in a black shirt bouncing on a sidewalk



A man in a black pullover and blue cap notches and shoots an arrow.



A woman in a blue outfit belly dancing.



A young girl wearing pink puts on makeup using a brush and compact.



A woman in a pink shirt ironing a piece of clothing



A woman is sitting on the floor unwrapping present.



A man in a bathing suit attempts a backward dive into a swimming pool.

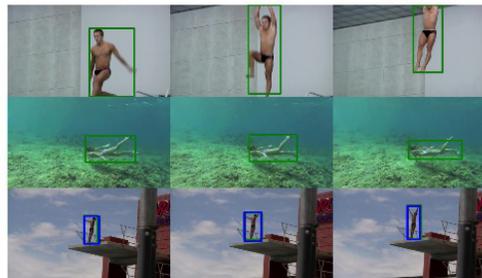
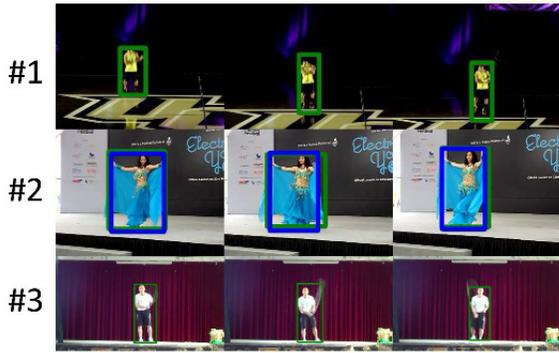
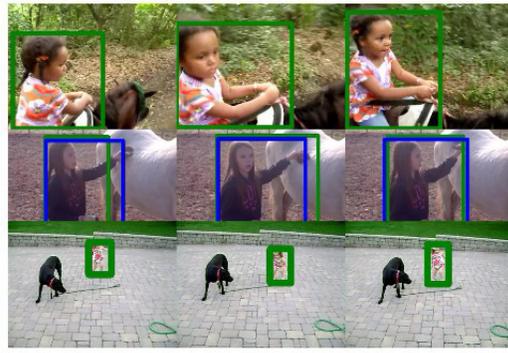


Figure 7. Randomly chosen examples. We show top-3 retrieved results for each description. The blue and green bounding boxes are the ground truth and the retrieved results, respectively.

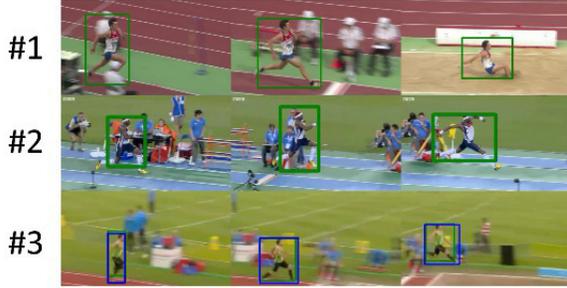
A woman in a blue dress dancing on the stage



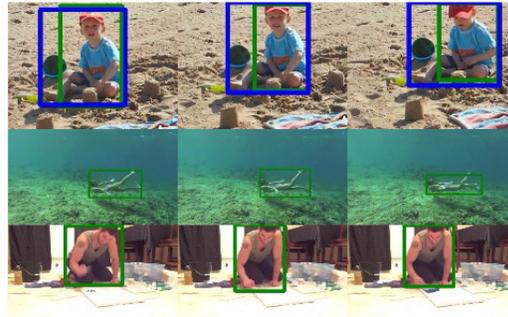
A young girl in a black jacket grooming a white horse with brushes



A young male Caucasian is performing a hurdle and long jump in slow motion.



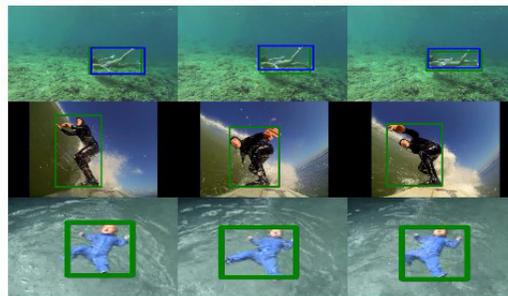
A young boy is making a castle in the sand.



The girl in the white shirt on the monkey bars



A beautiful women swimming under water with no air tank.



A young female demonstrates an illusion in a parking lot.



A man in black shorts and no shirt performing a pole vault.

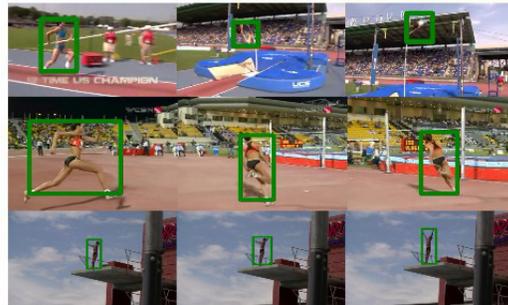


Figure 8. Randomly chosen examples. We show top-3 retrieved results for each description. The blue and green bounding boxes are the ground truth and the retrieved results, respectively.