Supplementary Material: Referring to Object in Video using Spatio-Temporal Identifying Description

Peratham Wiriyathammabhum[♠♢], Abhinav Shrivastava[♠♢], Vlad I. Morariu[♢], Larry S. Davis[♠♢]
University of Maryland: Department of Computer Science♠, UMIACS♢
peratham@cs.umd.edu, abhinav@cs.umd.edu
morariu@umd.edu, lsd@umiacs.umd.edu

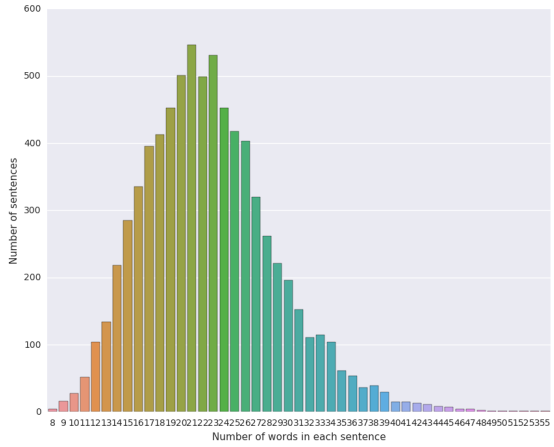


Figure 1: The number of words in a sentence in *STV-IDL* is normally distributed with an average of *22.65* words.

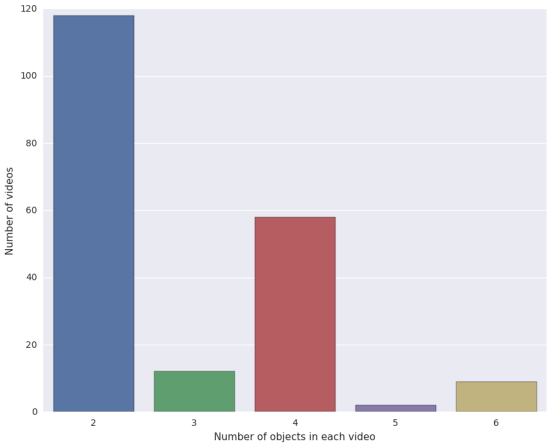


Figure 2: The number of target objects in *STV-IDL* is at least *2* with an average number of objects per video of *2.85*.

A Dataset Statistics

The length of referring expressions. We first analyze the length of referring expressions. We split each referring expression into words using Natural Language Toolkit (NLTK) (Bird et al., 2009). Figure 1 shows the distribution of the number of words in each referring expression. *STV-IDL* contains varying lengths of referring expressions and the average length (*22.65* words) is longer than most existing video referring expression datasets (Gao et al., 2017; Yamaguchi et al., 2017; Hendricks et al., 2017; Krishna et al., 2017) because we force the sentence syntax and encourage conjunctions.

The referring expressions provided from the annotators as speakers are subjective. However, as long as we correctly understand what they refer to and the process leads to mutual understanding and successful communication, the grounding process is valid. We are aware that giving an example sentence may limit the variations of the referring expressions. But, we want to make sure that our constraint is valid on every sentence and we also collect only a few referring expressions per referred object in a temporal interval. Besides, the collected sentences are usually from different annotators from the randomization in our web interface so there are still some constraint-satisfied variations within our few sentences per referred object. Compared to the ReferIt game (Kazemzadeh et al., 2014), the annotators are the speakers and we are the listeners. However, we are not collecting the data in a gamification setting in which referring expressions look short and concise similar to verbal utterances. The reason is it is not clear how a speaker will compare and utter words to contrast the referent from other distractors in the temporal dimension. Also, it is not clear how the listener will comprehend the referring expression in this setting for untrimmed video except rely on the time stamp to fast forward to the event itself. Our data collection pipeline is more similar to the Google-Refexp dataset (Mao et al., 2016) which

Table 1: Composition of the STV-IDL dataset, including number of videos and sentences for each collection.

Collection	Number of Videos	Number of Sentences
sepak_takraw	23	818
birds	24	426
dogs	18	410
elephants	11	301
panda	16	662
tennis_double	9	1160
tennis_single	15	668
tabletennis_double	9	336
tabletennis_single	10	376
badminton_double	12	816
badminton_single	18	408
beach_volley	14	812
fencing	20	372
STV-IDL	199	7569

has two separated steps, collecting the referring expression and then verification. This setting leads to grammatical sentences which are usually longer than verbal utterances.

The number of objects in each video. Figure 2 shows that our *STV-IDL* has multiple objects in each video with an average of *2.85* objects from the same class. From the annotation files, we observe that each video is annotated with *38.04* referring expressions on average.

Word occurrences. To ensure temporal words, Figure 8 shows that we have words like 'while', 'then', 'moves', 'steps', 'turns' in the top *50* frequency list. We further filter out stopwords in Figure 9. We can see that top words are words describing person (*man* or *player*) or appearances (*wearing* or *shirt* or *red*) or actions (*moves* or *steps*) or spatial relations (*front* or *right* or *left* or *back*). Temporal words like *then* are in NLTK stopword list.

Part-of-speech occurrences. To ensure the sentence syntax, Figure 10 shows the proportions of Penn part-of-speech tags (Taylor et al., 2003) using NLTK POS tagger. We observe that *STV-IDL* has a high proportion of prepositions (10.9 % IN), adjectives (9.7 % JJ), conjunctions (3.4 % CC) and adverbs (3.6 % RB) with only 28.1 % nouns (NN).

B Annotation Interfaces

We show good referring expression examples [Link] to the Amazon Mechanical Turk workers. We show a video clip with a green bounding box surrounding the target object. We then ask the annotators to write a sentence that contains at least *3 phrases*, a noun phrase, a verb phrase and a preposition or adverb phrase. A noun phrase should describe color or appearance. A verb phrase should describe actions. A preposition or adverb phrase should describe relations of the target object to other objects in the scene. We also encourage the annotators to write many phrases using conjunctions.

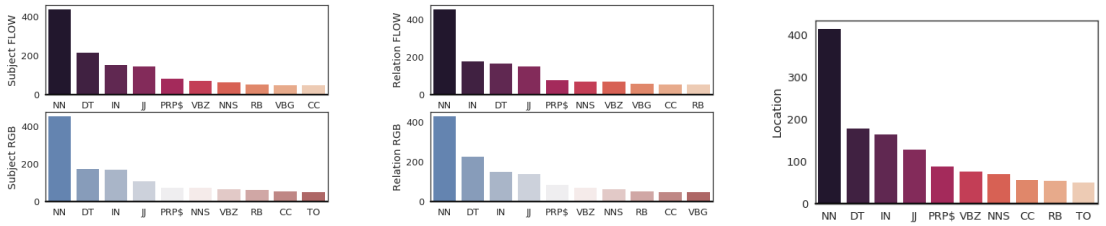
In the second stage, we take the annotated referring expressions to the verification interface. We manually verify that the sentence refers to the target object. If the sentence is valid, we then correct minor grammatical errors and misspelling. We paid workers 0.05$ to 0.10$ for each valid referring expression.

C Implementation Details

We implement our modular attention network based on PyTorch. The temporal interval proposal is implemented in Tensorflow. Optical flow images are estimated using the TVL-1 algorithm in OpenCV. For Faster-RCNN, we use a batch size of 8 and train on 4 GPUs for 20 epochs. The motion Faster-RCNN uses $128, 128, 128$ as the mean data value. For the modular attention network, we use ADAM optimizer with an initial learning rate $1e-3$, and we train for 30 epochs where we observed the learning process saturated. We use a 1-layer bidirectional LSTM for every LSTM in the model. The hidden layer of LSTM in language attention network is 512. For moving location module is 50. For relationship motion module is 20. All other settings are the same as the original implementation (Yu et al., 2018). We train the model using a combination of the ranking loss, the subject attribute loss, and the subject motion attribute loss. The training time takes three days for one stream models and a week for two-stream models on an NVIDIA P6000 GPU with 48GBs memory. We split the STV-IDL dataset into train, validation and test sets.

D Module Definitions

The language attention module learns to attend to each word for each visual module individually.



(a) Subject module RGB (below) and flow (top).

(b) Relationship module RGB (below) and flow (top).

(c) Location module. (coordinate + diffea)

Figure 3: Aggregations of output word attention weights for each module on the STV-IDL test set.

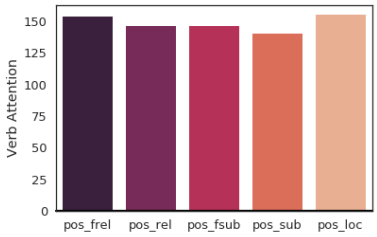


Figure 4: Aggregations on all verbs for each module. (from left to right: Relationship flow/RGB, Subject flow/RGB, Location)

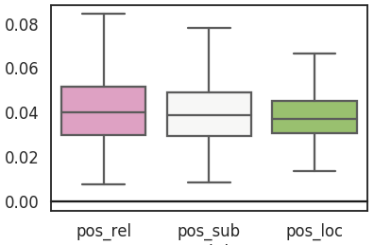


Figure 5: Aggregations on all verbs for each module. (from left to right: Relationship RGB, Subject RGB, and Location for RGB MAttNet) The model grounds more verbs to Subject RGB than fused1.

First, each word in the input expression is encoded into a one-hot vector. Then, a bidirectional LSTM encodes the whole expression. The hidden states in both direction for each time step are concatenated to create the word representation for each word. Next, a weighted vector is placed for each word representation and the weighted sum is the attention for each word. The phrase embedding is the weighted average of the one-hot vector and the attention for every word in the expression.

The subject module has two branches, attribute prediction and phase-guided attention pooling. In attribute prediction, the input expression is parsed to R1-R7 attributes (Kazemzadeh et al., 2014) using Stanford dependency parser (Manning et al., 2014). The attributes describe color, size, location and observed attribute types. Both 'pool5' and 'spatial_fc7' are concatenated and followed by a 1×1 convolution. The attribute feature blob is average-pooled and used for prediction by a fully connected layer. This branch is trained with a cross-entropy loss when the system can parse any attribute from the input expression. In phase-guided attention pooling, the attribute feature blob is concatenated with 'spatial_fc7' and phrase embedding followed by another 1×1 convolution to create the subject feature blob. The subject feature blob contains grids that correspond to the input image based on each spatial location. Then, the spatial attention is computed by forwarding the grid feature to tanh and softmax layers respectively. The weighted attention subject feature is computed by a weighted sum on the concatenation of attribute blob and 'spatial_fc7' with the spatial attention.

The subject module score is computed by matching weighted attention subject feature with the phrase embedding. The matching function consists of two MLPs with ReLU activations to project both features into a joint embedding space, two L2 normalization layers and a cosine similarity that measures the module score.

For the location module, the normalized bounding box coordinates, $l_i = [\frac{x_{min}}{W}, \frac{y_{min}}{H}, \frac{x_{max}}{W}, \frac{y_{max}}{H}, \frac{Area_{region}}{Area_{image}}]$ where W and H are width and height of the image, are computed for every object from the same class. Then, the location difference feature of the target object with up to five context objects from the same class, $\delta_{ij} = [\frac{\Delta x_{min}}{W}, \frac{\Delta y_{min}}{H}, \frac{\Delta x_{max}}{W}, \frac{\Delta y_{max}}{H}, \frac{\Delta Area_{region}}{Area_{image}}]$, is concatenated with the normalized bounding box coordinates $[l_i; \delta_{ij}]$ and fed to a fully connected layer which results in the final location feature.

The module score is from the matching between the final location feature and the phrase embedding using the matching function like in the subject module.

For the relationship module, the 'spatial_fc7' for top five context objects are averaged-pooled and become the context 'fc7' features. The 'fc7' feature is concatenated with the location difference feature $[\delta_{ij}]$ as the final relationship feature. However, the module score is from the maximum of the matching between the final relationship feature of each context object and the phrase embedding using the matching function like in the subject and location module. This is like putting all context objects into a bag in the weakly-supervised Multiple Instance Learning (MIL) setting and takes the best matching context as the final score.

For the motion stream, we train the motion Faster-RCNN on optical flow images of STV-IDL. We start from an MS COCO pretrained model, and we replace the three channel RGB input with a stack of flow-x, flow-y and flow magnitude from the flow image. Then, we duplicate the subject and the relationship module into the subject motion and relationship motion module which take the 'pool5' and 'spatial_fc7' features extracted from the motion Faster-RCNN.

Previous work (Simonyan and Zisserman, 2014) has shown that stacking many optical flow images can help recognition. So, we train another variant of two-stream modular attention network using stacked five optical flow frames. In this setting, we train the stacked motion Faster-RCNN by stacking flow images F_{idx} where frame index $idx \in [t-2, t+2]$. The input becomes a 15 channel stacked optical flow image. In addition, we add the moving location module to further model the movement of the location by stacking location features so that we have a sequence of $[l_i; \delta_{ij}]_{idx}$ where frame index $idx \in [t-2, t+2]$. Then, we place an LSTM on top of the sequence and we forward the concatenation of all hidden states to a fully connected layer and output the final location features. We also make a location sequence and placing an LSTM on top for location in the relationship motion module in this stacked optical flow setting. To fuse all modules, all module scores are weighted average by the language attention module.

E Experiments

The detailed experimental results are in Table 2. There are 3.37 objects per instance on average in the test set so randomly selecting one tubelet will get the accuracy of only 29.68% over 11% less than the baseline. The aggregated statistics of word attention are shown in Figure 3 and 4.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *International Conference on Computer Vision (ICCV)*.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In *International Conference on Computer Vision (ICCV)*.

Justin Johnson. simple-amt a microframework for working with amazon's mechanical turk. https://github.com/jcjohnson/simple-amt.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara L Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, pages 787–798.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *International Conference on Computer Vision (ICCV)*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20.

Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576.

Ann Taylor, Mitchell Marcus, and Beatrice Santorini. 2003. The penn treebank: an overview. In *Treebanks*, pages 5–22. Springer.

Table 2: Identifying Description Localization: mAP for each collection. (values are in percents.) The fused1 MAttNet is the proposed two-stream method and the fused5 MAttNet is the stacked version of the proposed two-stream method.

Collection	RGB MAttNet	flow MAttNet	flow5 MAttNet	fused1 MAttNet	fused5 MAttNet
sepak_takraw	**33.70**	14.67	30.43	27.17	17.93
birds	48.53	45.59	51.47	57.35	**64.71**
dogs	55.77	**61.54**	**61.54**	51.92	53.85
elephants	49.35	38.96	45.45	**63.64**	51.95
panda	52.34	53.27	**56.07**	42.06	51.40
tennis_double	18.38	**27.21**	23.53	20.59	26.47
tennis_single	**79.81**	55.77	64.42	71.15	62.50
tabletennis_double	25.83	32.50	26.67	28.33	**39.17**
tabletennis_single	33.33	51.67	54.17	**62.50**	45.83
badminton_double	36.57	32.87	33.33	**43.98**	35.65
badminton_single	64.47	75.00	81.58	**85.53**	81.58
beach_volley	34.41	25.81	28.49	31.72	**39.25**
fencing	55.70	**58.23**	48.10	51.90	48.10
STV-IDL	41.51	39.02	41.90	**44.66**	42.82

Masataka Yamaguchi, Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. 2017. Spatio-temporal person retrieval via natural language queries. In *International Conference on Computer Vision (ICCV)*.

Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. 2018. Mattnet: Modular attention network for referring expression comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Figure 6: The referring expression annotation interface based on a modified simple-amt (Johnson).

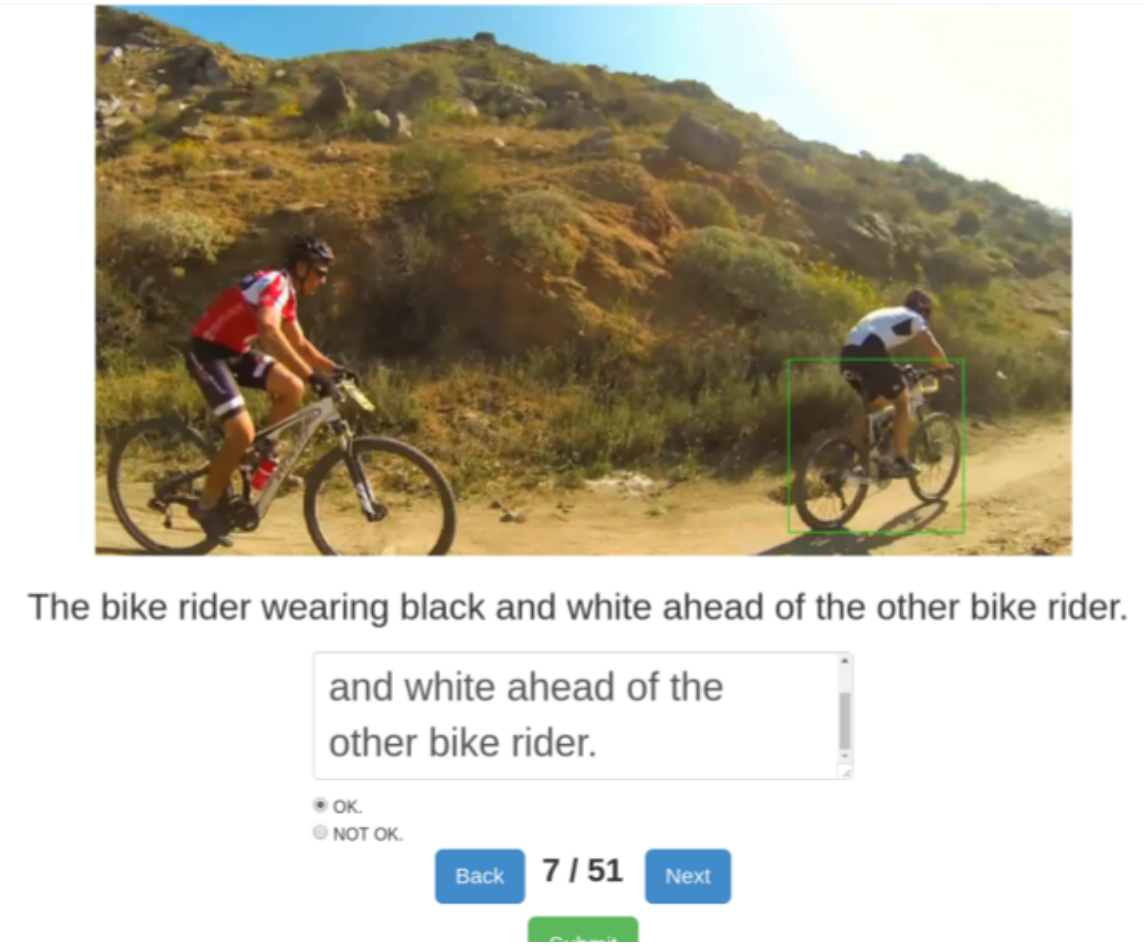


Figure 7: The referring expression verification interface allows the verifier to correct by editing the referring expression. Most grammatical errors and misspellings are corrected using this interface.

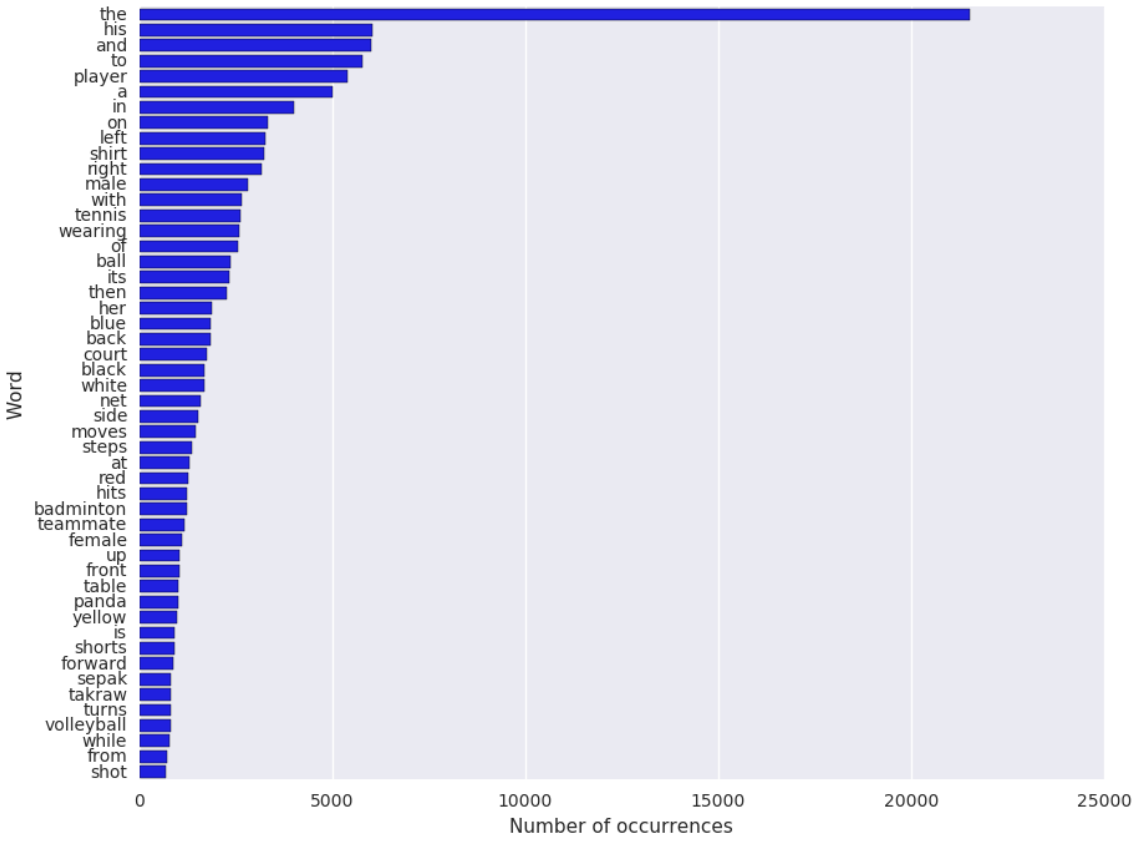


Figure 8: Top *50* words in *STV-IDL* with stopwords.

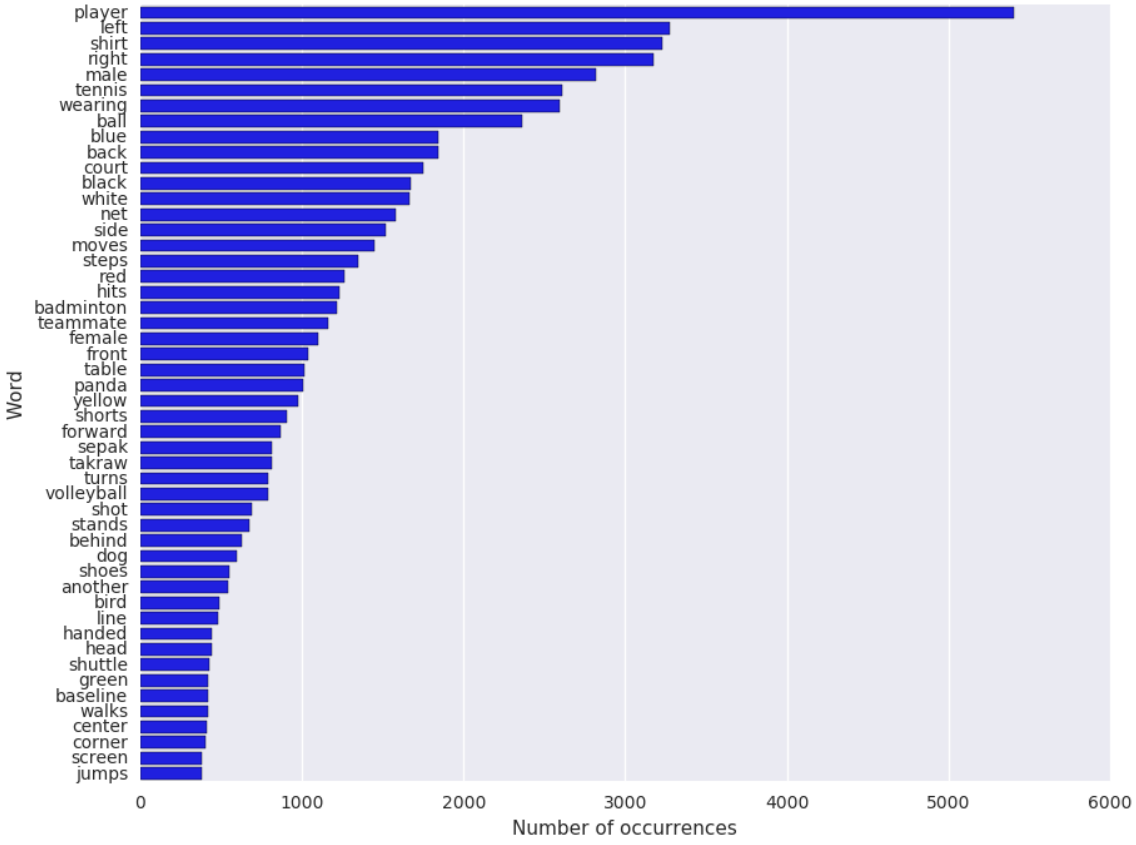


Figure 9: Top *50* words in *STV-IDL* without stopwords.

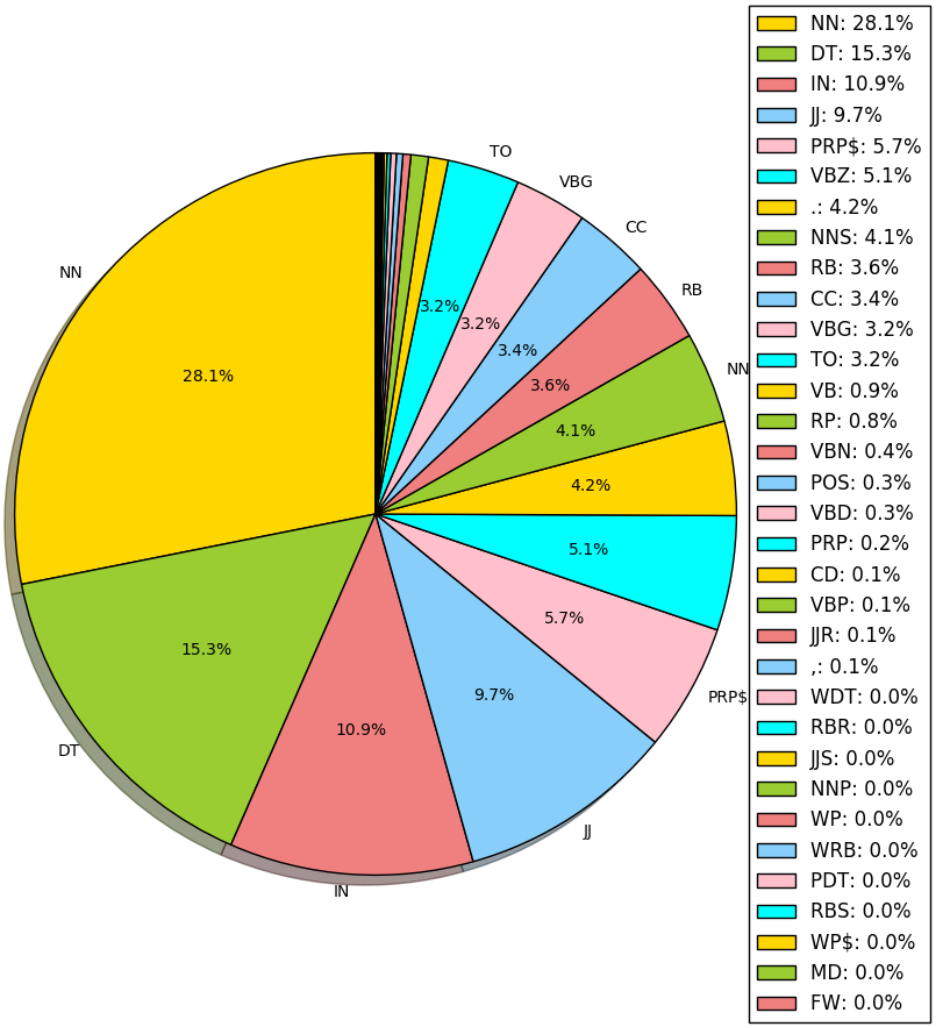


Figure 10: The percentages of each Part-of-speech in *STV-IDL*.

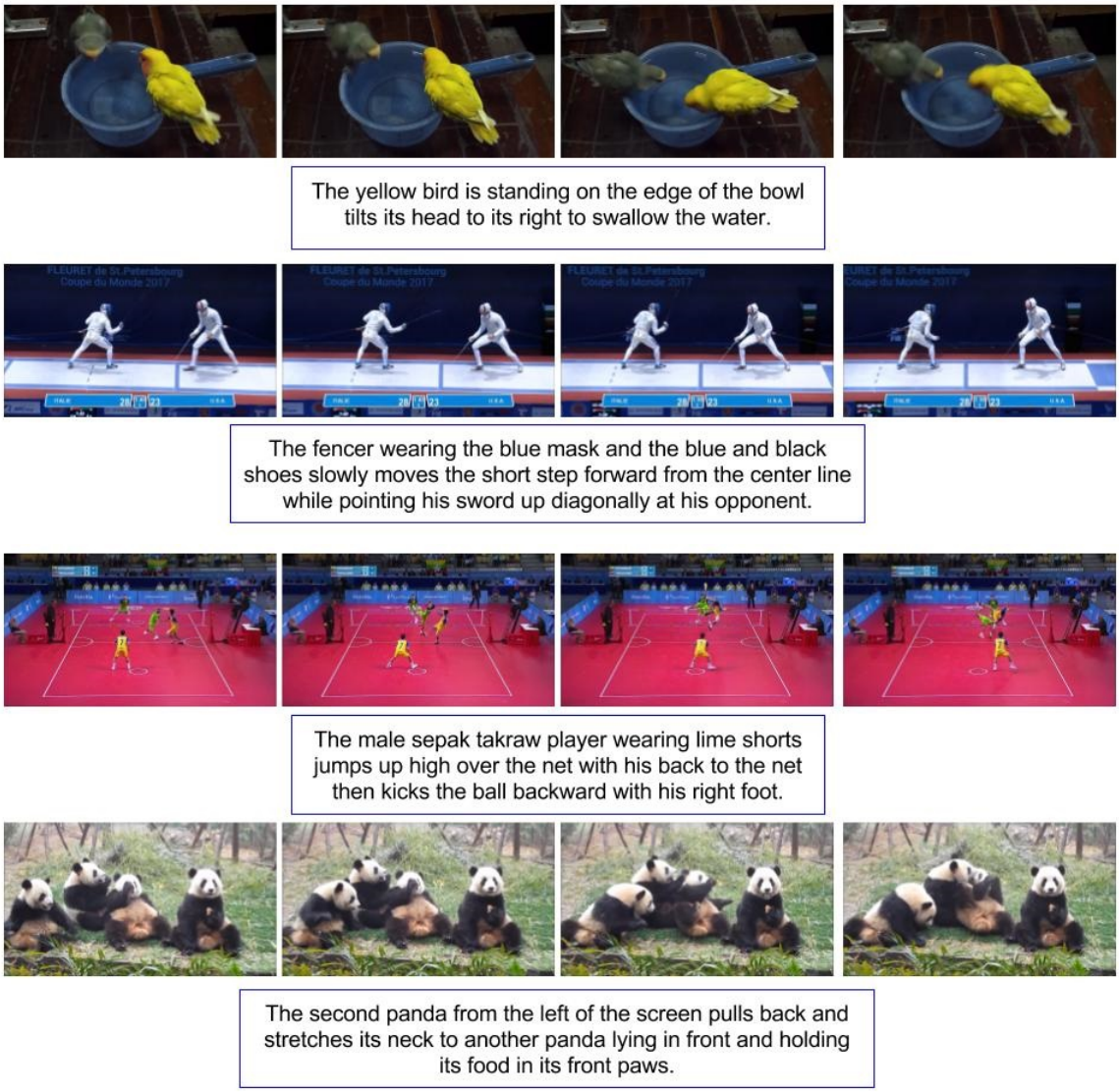


Figure 11: Sample data from *birds*, *fencing*, *sepak takraw* and *panda* collections.