Telling Stories for Common Sense Zero-Shot Action Recognition - Supplementary Material

Shreyank N Gowda[1] and Laura Sevilla-Lara[2]

[1] University of Nottingham, UK
[2] University of Edinburgh, UK
Shreyank.Narayanagowda@nottingham.ac.uk

We present experimental results for some our choices and discuss implementation details in the supplementary material. Some of our experimental choices included selecting the number of nearest neighbors for the data-based noise and the ranking loss. We discuss hyperparameter selection in Section 1, the need for cleaning data manually in Section 2 and the implementation details in Section 3.

1 Hyperparameter Selection

Choosing the number of nearest neighbors for both the data-based noise and the ranking loss is done empirically. We use the generalized zero-shot action recognition performance to decide these hyperparameters.

We choose UCF101 as our dataset for the hyperparameter tuning, but also plot the results on HMDB51 as it also ended up following the same pattern. The results are shown in Figure 1. Based on these results, we choose the number of nearest neighbors as 3 for the data-based noise and 5 for the ranking loss. The results reported are on the TruZe split.

2 Manual Cleaning

Generally, training with more data tends to produce better results. There often is a tension between using a smaller amount of clean data or a larger amount of noisy data. Here we have explored the effect of cleaning the data of Stories manually. In order to truly evaluate the effect of the Stories dataset, we evaluate multiple models on the noisy version of the Stories dataset and report results in Fig. 2. We see that using the noisy version of the dataset improves the performance over ER across methods but is still consistently worse than the cleaned version, even though it is roughly twice as large. This shows that the effect of cleaning up the data manually is not trivial.

3 Implementation Details

3.1 Ranking Loss

One of the risks of learning to generate semantic embeddings (through the “Projection Network" in is that synthetic semantic embeddings can be too similar to

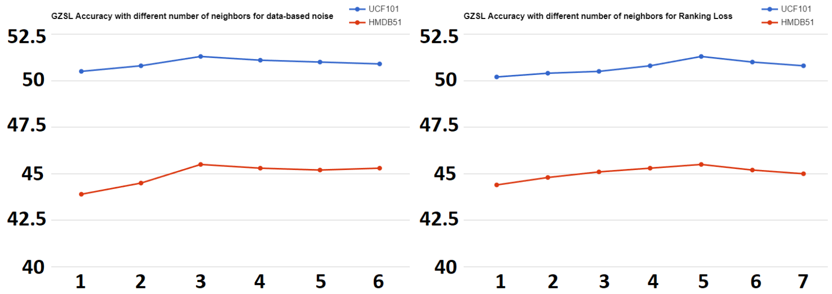

Fig. 1: Comparison of using different number of nearest neighbors on both (left) the data-based noise and (right) the ranking loss.

each other. To avoid this, we introduce a ranking loss [4] that pushes apart the generated semantic representation ($\hat{a}_i$) from those of their neighboring classes:

$$\mathcal{L}_{rank} = \mathbb{E}[max(0, \delta - a^T \hat{a}_i + (a')^T \hat{a}_i)], \tag{1}$$

where a is the ground truth semantic embedding, a' is the semantic embedding of a class randomly sampled from the 5 classes (empirical results in Sec. 1 closest to the ground truth and δ is a hyperparameter. Including this loss in the overall objective function, we obtain:

$$\begin{aligned} &\min_{G} \min_{P} \max_{D} \mathcal{L}_{\mathcal{D}} + \lambda_1 \mathcal{L}_{CLS}(G) \\ &+\lambda_2 \mathcal{L}_{rank}(P) + \lambda_3 \mathcal{L}_{MI}(G). \end{aligned} \tag{2}$$

3.2 Features

For our visual features we consider two scenarios. The first case, the appearance and flow features are extracted from the *Mixed 5c* layer of the RGB and flow I3D networks, respectively. Both I3D models are pre-trained on the Kinetics-400 dataset [2].

Given an input video, appearance and flow features extracted are averaged across the temporal dimension and pooled by 4 in the spatial dimension and then flattened to obtain a vector of size 4096 each. These vectors are then concatenated to obtain video features of size 8192.

In the second case, we first train the X-CLIP-B/16 [9] on 16 frames of the non-overlapping classes of Kinetics [1] dubbed Kinetics-664 [1] using the proposed 'Stories' as the semantic embedding. For the text embeddings we use the large S-BERT [10], which is a sentence encoder.

For ER we use the class definition as input to the S-BERT and use the 1024 sized vector output as the semantic embedding. In case of Stories, we use S-BERT for each sentence and average all the vectors to obtain a singe vector of size 1024.

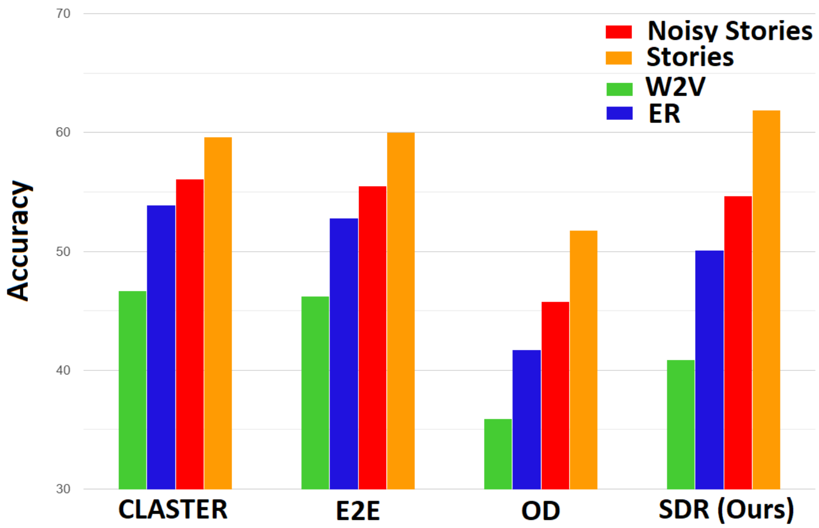

Fig. 2: Using the cleaned version of *Stories* to create the semantic space of class labels improves the performance by a large margin. The dataset is UCF101.

3.3 Network Architecture

We use the Wasserstein GAN [11] which has been successful in both zero-shot image classification [12] and zero-shot action recognition [8] tasks. This also allows us to compare directly to OD [8] and Wasserstein GAN [11] in the experimental analysis.

The feature generator G is a three-layer fully-connected network that has an output layer dimension equal to that of the video feature size. The hidden layers are of size 4096. The discriminator D is also a three-layer fully-connected network with hidden layers of size 4096. However, the output size equals 1. The projection network P is a fully-connected network that has an output layer size equal to the size of the semantic embeddings (in our case 1024).

3.4 Training Details

All the modules are trained using the Adam optimizer with a weight decay of 0.0005 and with an adaptive learning rate using a learning rate scheduler. We set λ_1 as 0.1, λ_2 as 0.9 and λ_3 as 0.1. At test time, we follow OD [8] and train a single classifier for ZSAR and two classifiers for GZSAR along with an out-of-distribution (OOD) detector. We report on TruZe [7] to avoid any inflated accuracies due to class overlap.

The classifiers are single-layer fully-connected networks with an input size equal to the video feature size and output sizes equal to the number of classes (seen or unseen). The OOD is a three-layer fully connected network with output and hidden layer sizes equal to the number of seen classes and 512, respectively. We use 8 RTX 2080 Ti NVIDIA GPUs having 16 GB RAM each for our experiments.

4 Why Not Just Use VAE for Feature Generation?

Another possible question is the use of the current feature generator model. There are multiple options to use as feature generators including VAEs, and other versions of GANs (not just WGAN [11] that we use).

We chose to adapt the WGAN for our feature generator based on two reasons. First, we wanted to compare directly to existing literature on zero-shot action recognition and to the best of our knowledge the most recent one has been the one used in OD [8].

However, for the sake of sanity we also ran additional experiments on the HMDB51 dataset incorporating f-VAEGAN [13], adapted FREE [3]: feature refinement of f-VAEGAN for zero-shot action recognition and using a simple VAE. The results of this can be seen in Tab 1.

Feature Generator	Accuracy
VAE	25.5 ± 2.9
Vanilla GAN	31.5 ± 2.4
f-VAEGAN	45.9 ± 3.2
FREE	46.6 ± 3.5
SDR (Ours)	**48.1 ± 3.6**

Table 1: Comparing different choices for feature generator. Reported results are on 10 different runs and all models use the same split. Dataset is HMDB51.

5 Generalized Zero-Shot Action Recognition Results in Detail

In order to better analyze performance of the model on GZSL, we report the average seen and unseen accuracies along with their harmonic mean. The results using different embeddings and on the UCF101, HMDB51 and Olympics datasets are reported in Table 2.

The reported results are on the same set of 10 random splits for fair compairson. There are no manual attributes for the HMDB dataset. We see that the proposed SDR approach obtains best results on all three categories. Another observation we can see is that the performance of all models using *Stories* is better than even the older manual attributes.

Model	SE	Olympics			HMDB51			UCF-101		
		u	s	H	u	s	H	u	s	H
WGAN [11]	M	50.8	71.4	59.4	-	-	-	30.4	**83.6**	44.6
OD [8]	M	61.8	71.1	66.1	-	-	-	36.2	76.1	49.1
SPOT [5]	M	-	-	69.1	-	-	-	-	-	51.8
CLASTER [6]	M	66.2	71.7	68.8	-	-	-	40.2	69.4	50.9
SDR	M	**71.6**	**76.9**	**74.2**	-	-	-	**43.1**	77.5	**54.6**
WGAN [11]	W	35.4	65.6	46.0	23.1	55.1	32.5	20.6	73.9	32.2
OD [8]	W	41.3	72.5	52.6	25.9	55.8	35.4	25.3	74.1	37.7
CLASTER	W	49.2	71.1	58.1	35.5	52.8	42.4	30.4	68.9	42.1
WGAN [11]	S	36.1	66.2	46.7	28.6	57.8	38.2	27.5	74.7	40.2
OD [8]	S	42.9	73.5	54.1	33.4	57.8	42.3	32.7	75.9	45.7
CLASTER	S	49.9	71.3	58.7	42.7	53.2	47.4	36.9	69.8	48.3
CLASTER	C	66.8	71.6	69.1	43.7	53.3	48.0	40.8	69.3	51.3
WGAN [11]	Sto	52.5	73.4	61.2	35.2	65.1	45.7	33.8	**84.2**	48.2
OD [8]	Sto	63.3	75.1	68.7	37.2	**67.5**	47.9	40.1	81.7	53.8
CLASTER	Sto	69.1	74.1	71.5	44.3	57.2	49.9	42.1	71.5	53.0
SDR+I3D	Sto	73.5	79.9	76.6	46.9	55.8	50.9	44.4	80.7	57.2
SDR+CLIP	Sto	**78.9**	**83.5**	**81.1**	**52.5**	60.4	**56.1**	**47.3**	81.2	**59.7**

Table 2: Seen and unseen accuracies for CLASTER on different datasets using different embeddings. 'SE' corresponds to the type of embedding used, wherein 'M', 'W', 'S', 'C' and 'Sto' refers to manual annotations, word2vec, sen2vec, combination of the embeddings and Stories respectively. 'u', 's' and 'H' corresponds to average unseen accuracy, average seen accuracy and the harmonic mean of the two. All the reported results are on the same splits. SDR+I3D corresponds to the backbone network being I3D and similarly for SDR+CLIP.

References

1. B. Brattoli, J. Tighe, F. Zhdanov, P. Perona, and K. Chalupka. Rethinking zero-shot video classification: End-to-end training for realistic applications. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4613–4623, 2020. 2
2. J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017. 2
3. S. Chen, W. Wang, B. Xia, Q. Peng, X. You, F. Zheng, and L. Shao. Free: Feature refinement for generalized zero-shot learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 122–131, 2021. 4
4. A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems*, 26, 2013. 2
5. S. N. Gowda. Synthetic sample selection for generalized zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 58–67, 2023. 5
6. S. N. Gowda, L. Sevilla-Lara, F. Keller, and M. Rohrbach. Claster: clustering with reinforcement learning for zero-shot action recognition. In *European Conference on Computer Vision*, pages 187–203. Springer, 2022. 5

7. S. N. Gowda, L. Sevilla-Lara, K. Kim, F. Keller, and M. Rohrbach. A new split for evaluating true zero-shot action recognition. *arXiv preprint arXiv:2107.13029*, 2021. 3
8. D. Mandal, S. Narayan, S. K. Dwivedi, V. Gupta, S. Ahmed, F. S. Khan, and L. Shao. Out-of-distribution detection for generalized zero-shot action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9985–9993, 2019. 3, 4, 5
9. B. Ni, H. Peng, M. Chen, S. Zhang, G. Meng, J. Fu, S. Xiang, and H. Ling. Expanding language-image pretrained models for general video recognition. In *European Conference on Computer Vision*, pages 1–18. Springer, 2022. 2
10. N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. 2
11. Y. Xian, T. Lorenz, B. Schiele, and Z. Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5542–5551, 2018. 3, 4, 5
12. Y. Xian, B. Schiele, and Z. Akata. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4582–4591, 2017. 3
13. Y. Xian, S. Sharma, B. Schiele, and Z. Akata. f-vaegan-d2: A feature generating framework for any-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10275–10284, 2019. 4