

# Practical Aspects of Zero-Shot Learning

Elie Saad<sup>1</sup>[0000–0002–5627–6351], Marcin Paprzycki<sup>2</sup>[0000–0002–8069–2152], and  
Maria Ganzha<sup>1</sup>[0000–0001–7714–4844]

<sup>1</sup> Warsaw University of Technology, Warsaw, Poland  
elie.saad.stud@pw.edu.pl, maria.ganzha@pw.edu.pl

<sup>2</sup> Systems Research Institute Polish Academy of Sciences  
marcin.paprzycki@ibspan.waw.pl

**Abstract.** Zero-shot learning is applied, for instance, when properly labeled training data is not available. A number of zero-shot algorithms have been proposed. However, since none of them seems to be an “overall winner”, development of a meta-classifier(s) combining “best aspects” of individual classifiers can be attempted. In this context, state-of-the-art zero-shot learning methods are compared for standard benchmark datasets. Next, multiple meta-classifiers are applied to the same datasets.

**Keywords:** Zero-Shot Learning · Meta-Classifer · Benchmarking.

## 1 Introduction and Literature Review

Many real-world applications require classifying “entities” not encountered earlier, e.g., object recognition (where every object is a category), cross-lingual dictionary induction (where every word is a category), etc. Here, one of the reasons is lack of resources to annotate available (and possibly systematically growing) datasets. To solve this problem, zero-shot learning has been proposed.

While multiple zero-shot learning approaches have been proposed ([9,19]), as of today, none of them emerged as “the best”. In situations like this, meta-classifiers, which “receive suggestions” from individual classifiers and “judge” their value to select a “winner”, can be proposed. The assumption is that such meta-classifier will perform better than the individual ones.

Let us start from the formal *problem formulation*. Given a dataset of *image embeddings*  $\mathcal{X} = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y} | i = 1, \dots, N_{tr} + N_{te}\}$ , each image is a real  $D$ -dimensional embedding vector comprised of features  $x_i \in \mathbb{R}^D$ , and each class label is represented by an integer  $y_i \in \mathcal{Y} \equiv \{1, \dots, N_0, N_0 + 1, \dots, N_0 + N_1\}$  giving  $N_0 + N_1$  distinct classes. Here, for generality, it is assumed that  $\mathcal{X} \stackrel{\text{def}}{=} \mathbb{R}^D$ . The dataset  $\mathcal{X}$  is divided into two subsets: (1) training set and (2) test set. The training set is given by  $X^{tr} = \{(x_i^{tr}, y_i^{tr}) \in \mathcal{X} \times \mathcal{Y}_0 | i = 1, \dots, N_{tr}\}$ , where  $y_i^{tr} \in \mathcal{Y}_0 \equiv \{1, \dots, N_0\}$ , resulting in  $N_0$  distinct training classes. The test set is given by  $X^{te} = \{(x_i^{te}, y_i^{te}) \in \mathcal{X} \times \mathcal{Y}_1 | i = N_{tr} + 1, \dots, N_{tr} + N_{te}\}$ , where  $y_i^{te} \in \mathcal{Y}_1 \equiv \{N_0 + 1, \dots, N_0 + N_1\}$  providing  $N_1$  distinct test classes.

The goal of zero-shot learning is to train a model (on dataset  $X^{tr}$ ) that performs “well” for the test dataset  $X^{te}$ . Obviously, zero-shot learning requires

auxiliary information associating labels from the training and the test sets, when  $\mathcal{Y}_0 \cap \mathcal{Y}_1 = \emptyset$ . The solution is to represent each class label  $y$  ( $1 \leq y \leq N_0 + N_1$ ) by its prototype  $\pi(y) = p \in \mathcal{P} \stackrel{\text{def}}{=} \mathbb{R}^M$  (semantic embedding). Here,  $\pi(\cdot) : \mathcal{Y}_0 \cup \mathcal{Y}_1 \rightarrow \mathcal{P}$  is the prototyping function, and  $\mathcal{P}$  is the semantic embedding space. The prototype vectors are such that any two class labels  $y_0$  and  $y_1$  are similar if and only if their prototype representations  $\pi(y_0) = p_0$  and  $\pi(y_1) = p_1$  are close in the semantic embedding space  $\mathcal{P}$ . For example, their inner product is large in  $\mathcal{P}$ , i.e.,  $\langle \pi(y_0), \pi(y_1) \rangle_{\mathcal{P}}$  is large. Prototyping all class labels into a joint semantic space, i.e.,  $\{\pi(y) | y \in \mathcal{Y}_0 \cup \mathcal{Y}_1\}$ , results in labels becoming related. This resolves the problem of disjoint class sets, and the model can learn from the labels in the training set, and predict labels from the test set.

Multiple algorithms have been proposed to solve the zero-shot learning problem. Here, *DeViSE* [6], *ALE* [2], and *SJE* [3] use a bilinear compatibility function. They follow the *Stochastic Gradient Descent* (SGD), implicitly regularized by early stopping. The *ESZSL* [12] uses square loss to learn the bilinear compatibility function, and explicitly defines regularization with respect to the Frobenius norm. Kodirov et al. in [8] proposes a semantic encoder-decoder model (*SAE*), where the training instances are projected into the semantic embedding space  $\mathcal{P}$ , with the projection matrix  $W$ , and then projected back into the feature space  $\mathcal{X}$ , with the conjugate transpose of the projection matrix  $W^*$ . Another group of approaches adds a non-linearity component to the linear compatibility function [18]. Third set of approaches uses probabilistic mappings [9]. Fourth group of algorithms expresses the input image features and the semantic embeddings as a mixture of seen classes [21]. In the fifth approach, both seen and unseen classes are included in the training data [20].

In this context, a comparison of five state-of-the-art zero-shot learning approaches, applied to five popular benchmarking datasets, is presented. Next, explorations into meta-classifier for zero-shot learning are reported. Extended version of this work, with additional details and results, can be found in [14].

## 2 Selection of Methods and Experimental Setup

Based on the analysis of the literature, five robust zero-shot learning approaches were selected: (1) *DeViSE*, (2) *ALE*, (3) *SJE*, (4) *ESZSL*, and (5) *SAE*. Moreover, the following, popular in the literature, datasets have been picked: (a) *CUB* [17], (b) *AWA1* [9], (c) *AWA2* [19], (d) *aPY* [5], and (e) *SUN* [11]. Finally, five standard meta-classifiers have been tried: (A) Meta-Decision Tree *MDT* [16], (B) deep neural network *DNN* [7], (C) Game Theory-based approach *GT* [1], (D) Auction-based model *Auc* [1], (E) Consensus-based approach *Con* [4], and (F) a simple majority voting *MV* [13]. Here, classifiers (C), (D), (E) and (F) have been implemented following cited literature. However, the implementation of (A) differs from the one described in [16] by not applying the *weight* condition on the classifiers. However, effects of this simplification can be explored in the future. Finally, the *DNN* has two hidden layers and an output layer. All of them use the rectified linear activation function. The optimization function is the

SGD, with the mean squared error loss function. All codes and complete list of hyperparameter values for the individual classifiers and the meta-classifiers can be found in the Github repository<sup>3</sup>. While hyperparameter values, were obtained through multiple experiments, no claim about their optimality is made. The following standard measures have been used to measure the performance of the explored approaches: (T1) Top-1, (T5) Top-5, (LogLoss) Logarithmic Loss, and (F1) F1 score. Their definitions can be found in [19,10,15].

Separately, comparison with results reported in [19] has to be addressed. To the best of our knowledge, codes used there are not publicly available. Thus, the best effort was made to re-implement methods from [19]. As this stage, the known differences are: (1) feature vectors and semantic embedding vectors, provided in the datasets were used, instead of the calculated ones; (2) dataset splits for the written code follow the splits proposed in [19], instead of the “standard splits”. Nevertheless, we believe that the results, presented herein, fairly represent the work reported in [19].

### 3 Experiments with Individual Classifiers

The first set of experimental results was obtained using the five classifiers applied to the five benchmark datasets. Results displayed in Table 1 show those available from [6,2,3,12,8] (in the **O** column). The **R** column represents results based on [19]. The **I** columns represents the in-house implementations of the five models. Overall, all classifiers performed “badly” when applied to the *aPY* dataset.

**Table 1.** Individual Classifier Performance for the Top-1 Accuracy

CLF	CUB			AWA1			AWA2		aPY			SUN		
	O	R	I	O	R	I	R	I	O	R	I	O	R	I
DeViSE	-	52	46.82	-	54.2	53.97	59.7	57.43	-	37	32.55	-	56.5	55.42
ALE	26.3	54.9	<b>56.34</b>	47.9	59.9	56.34	62.5	51.89	-	39.7	33.4	-	58.1	<b>62.01</b>
SJE	50.1	53.9	49.17	66.7	65.6	<b>58.63</b>	61.9	<b>59.88</b>	-	31.7	31.32	-	52.7	52.64
ESZSL	-	51.9	53.91	49.3	58.2	56.19	58.6	54.5	15.1	38.3	<b>38.48</b>	65.8	54.5	55.63
SAE	-	33.3	39.13	84.7	53.0	51.5	54.1	51.77	-	8.3	15.92	-	40.3	52.71

Next, comparing the results between columns *R* and *I*, out of 25 results, methods based on [19] are somewhat more accurate in 15 cases. Hence, since performances are close, and one could claim that our implementation of methods from [19] is “questionable”, from here on, only results based on “in house” implementations of zero-shot learning algorithms are reported.

While the Top-1 performance measure is focused on the “key class”, other performance measures have been tried. In [14] performance measured using *Top 5*, *LogLoss*, and *F1 score* have been reported. Overall, it can be stated that (A) different performance measures “promote” different zero-shot learning approaches; (B) *aPY* is the “most difficult dataset” regardless of the measure; (C) no individual classifier is clear winner. Therefore, a simplistic method has been proposed, to

<sup>3</sup> [https://github.com/Inars/Developing\\_MC\\_for\\_ZSL](https://github.com/Inars/Developing_MC_for_ZSL)

gain a better understanding of the “overall strength” of each classifier. However, what follows “should be treated with a grain of salt”. Here, individual classifiers score points ranging from 5 to 1 (from best to worst) based on the accuracy obtained for each dataset. This process is applied to all four accuracy measures. Combined scores have been reported in Table 2. Interestingly, *SAE* is the weak-

**Table 2.** Individual Classifier Combined Performance; “winners” marked in bold font.

CLF	CUB	AWA1	AWA2	aPY	SUN	Total
DeViSE	8	8	17	16	11	60
ALE	<b>19</b>	11	7	<b>15</b>	<b>18</b>	70
SJE	12	<b>19</b>	<b>18</b>	<b>15</b>	8	72
ESZSL	17	14	14	11	17	<b>73</b>
SAE	4	8	8	7	6	33

est method for both the individual datasets and the overall performance. The combined performance of *ALE*, *SJE*, and *EZSL* is very similar.

### 3.1 Analysis of the Datasets

Since it became clear that the performance of the classifiers is directly related to the datasets, their “difficulty” has been explored. Hence, an instance (in a dataset) is classified as *lvl 0* if *no* classifier identified it correctly, whereas *lvl 5* if it was recognized by *all* classifiers. The results in Table 3, show how many instances (in %) belong to each category, for each dataset. Here, the *aPY* dataset

**Table 3.** Analysis of Instance Difficulty (represented in %)

CLF	lvl 0	lvl 1	lvl 2	lvl 3	lvl 4	lvl 5
CUB	23.86	15.98	13.11	12.4	15.23	19.41
AWA1	20.47	15.67	11.29	12.19	<b>19.33</b>	21.04
AWA2	21.88	12.95	13.46	<b>14.62</b>	12.26	22.84
aPY	<b>36.37</b>	<b>25.76</b>	<b>15.75</b>	11.75	9.53	0.85
SUN	19.31	12.78	10.69	12.5	16.88	<b>27.85</b>

has the largest percent of instances that have not been recognized at all (36.37%), or by one or two classifiers (jointly, 41.51%). At the same time, only 0.85% of its instances have been recognized by all classifiers. The *SUN* dataset is the easiest: 27.85% of instances were correctly recognized by all classifiers and about 55% of its instances are “relatively easy”.

Approaching the issue from different perspective, the “influence” of individual attributes has been “scored”. For each correctly recognized instance, its attributes have been given +1 “points”. For incorrectly recognized instances, their attributes were given -1 “points”. This measure captured which attributes are the *easiest/hardest* to recognize. Obtained results can be found in Table 4. The most interesting observation is that attributes: “has eye color black” and “metal” are

**Table 4.** Analysis of the Datasets

CLF	Easiest Attribute	Individual Classifier	Hardest Attribute
CUB	has eye color black	all	has eye color black
AWA1	old world	DeViSE; SAE	group
	fast	ALE	ocean
	old world	SJE; ESZSL	ocean
	quadrupedal	total	swims
AWA2	old world	DeViSE; ALE; SJE; SAE; total	group
	old world	ESZSL	ocean
aPY	metal	DeViSE; ALE; SJE; ESZSL	metal
	head	SAE	metal
	furry	total	metal
SUN	no horizon	all	man-made

associated with so many instances that they are classified (in)correctly regardless if they actually “influenced” the “decision” of the classifier.

## 4 Meta-Classifiers

Let us now move to meta-classifiers. Here, note that the number of *hard* instances, found in each dataset (see, Section 2), establishes the hard ceiling for: *DNN*, *MDT*, and *MV*. Specifically, if not a single classifier gave a correct answer, in these approaches, their combination will also “fail”. In Table 5, the performance of the six meta-classifiers is compared using the Top-1 measure, where the **Best** row denotes the best result obtained by the “winning” individual classifier, for a given dataset (see, Table 1). Results using the F1 score, can be found in [14].

**Table 5.** Meta-classifier performance; Top-1 accuracy

CLF	CUB	AWA1	AWA2	aPY	SUN
MV	<b>53.43</b>	<b>58.71</b>	56.56	32.72	61.94
MDT	47.89	56.43	51.89	33.40	<b>62.08</b>
DNN	48.63	57.56	54.72	<b>34.89</b>	60.63
GT	46.58	56.75	<b>59</b>	32.63	59.51
Con	46.82	53.97	57.43	32.55	55.42
Auc	47.89	56.34	51.89	33.40	62.01
Best	56.34	58.63	59.88	38.48	62.01

Comparing the results, one can see that: (a) the best individual classifier performed better than the best meta-classifier on *CUB*, *AWA2*, and *aPY* (2.91%, 0.88%, and 3.59% better); (b) the best meta-classifier performed better than the best individual classifier on *AWA1* and *SUN* datasets (0.08% and 0.08% better).

Finally, the “scoring method” was applied jointly to the meta-classifiers and the individual classifiers, for the Top-1 and the F1 score accuracy measures. Obviously, since 11 classifiers were compared, the top score was 11 points. Table 6 displays the results. It can be noticed that (1) meta-classifiers performed better

**Table 6.** Meta-Classifier and individual classifier combined performance

CLF	CUB	AWA1	AWA2	aPY	SUN	Total
DeViSE	13	12	20	12	13	70
ALE	<b>21</b>	15	11	17	21	85
SJE	17	<b>20</b>	<b>22</b>	16	10	85
ESZSL	21	16	16	16	15	84
SAE	10	10	11	7	11	49
MV	20	21	17	<b>18</b>	20	<b>96</b>
MDT	15	16	11	<b>18</b>	<b>22</b>	82
DNN	17	18	15	17	18	85
GT	10	14	15	11	13	63
Con	13	10	13	12	13	61
Auc	15	15	11	<b>18</b>	21	80

than the individual classifiers (averaging 77.83 vs. 74.6 points). (2) Combining results from the individual classifiers using a simple *majority voting* algorithm brought best results. At the same time, (3) use of basic versions of more advanced meta-classifiers is not leading to immediate performance gains.

## 5 Concluding Remarks

In this contribution, performance of five zero-shot learning models has been studied, when applied to popular benchmarking datasets. Moreover, the “nature of difficulty” of these datasets has been explored. Finally, six standard meta-classifiers have been experimented with. The main findings were: (1) there is no single best classifier, and results depend on the dataset and the performance measure; (2) the *aPY* dataset is the most difficult for zero-shot learning; (3) standard meta-classifiers may bring some benefits; (4) the simplest methods obtained best results (i.e., the individual classifier *ESZSL* and the meta-classifier *MV*). The obtained prediction accuracy (less than 70%) suggests that a lot of research is needed for both the individual classifiers and, possibly, the meta-classifiers. Moreover, datasets similar to the *aPY*, which are particularly difficult achieve good performance, should be used. Finally, some attention should be devoted to the role that individual attributes play in class (instance) recognition.

**Acknowledgements** Research funded in part by the Centre for Priority Research Area Artificial Intelligence and Robotics of Warsaw University of Technology within the Excellence Initiative: Research University (IDUB) programme.

## References

1. Abreu, M.d.C., Canuto, A.M.: Analyzing the benefits of using a fuzzy-neuro model in the accuracy of the neurage system: An agent-based system for classification tasks. In: Proc. IEEE Int. Joint Conf. on NN. pp. 2959–2966. IEEE (2006)
2. Akata, Z., Perronnin, F., Harchaoui, Z., Schmid, C.: Label-embedding for image classification. IEEE transactions on pattern analysis and machine intelligence **38**(7), 1425–1438 (2015)

3. Akata, Z., Reed, S., Walter, D., Lee, H., Schiele, B.: Evaluation of output embeddings for fine-grained image classification. In: Proc. of the IEEE conf. on computer vision and pattern recognition. pp. 2927–2936 (2015)
4. Alzubi, O.A., Alzubi, J.A.A., Tedmori, S., Rashaideh, H., Almomani, O.: Consensus-based combining method for classifier ensembles. *Int. Arab J. Inf. Technol.* **15**(1), 76–86 (2018)
5. Farhadi, A., Endres, I., Hoiem, D., Forsyth, D.: Describing objects by their attributes. In: 2009 IEEE conf. on computer vision and pattern recognition. pp. 1778–1785. IEEE (2009)
6. Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Ranzato, M., Mikolov, T.: Devise: A deep visual-semantic embedding model. *Advances in neural inf. proc. sys.* **26** (2013)
7. Goodfellow, I., Bengio, Y., Courville, A.: *Deep learning*. MIT press (2016)
8. Kodirov, E., Xiang, T., Gong, S.: Semantic autoencoder for zero-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3174–3183 (2017)
9. Lampert, C., Nickisch, H., Harmeling, S.: Attribute-based classification for zero-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36**(3) (2013)
10. Mannor, S., Peleg, D., Rubinstein, R.: The cross entropy method for classification. In: Proc. of the 22nd int. conf. on ML. pp. 561–568 (2005)
11. Patterson, G., Hays, J.: Sun attribute database: Discovering, annotating, and recognizing scene attributes. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. pp. 2751–2758. IEEE (2012)
12. Romera-Paredes, B., Torr, P.: An embarrassingly simple approach to zero-shot learning. In: Int. conf. on ML. pp. 2152–2161. PMLR (2015)
13. Ruta, D., Gabrys, B.: Classifier selection for majority voting. *Information fusion* **6**(1), 63–81 (2005)
14. Saad, E., Paprzycki, M., Ganzha, M.: Practical aspects of zero-shot learning. <https://doi.org/10.48550/ARXIV.2203.15158>, <https://arxiv.org/abs/2203.15158>
15. Sokolova, M., Japkowicz, N., Szpakowicz, S.: Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In: Australasian joint conf. on AI. pp. 1015–1021. Springer (2006)
16. Todorovski, L., Džeroski, S.: Combining classifiers with meta decision trees. *Machine learning* **50**(3), 223–249 (2003)
17. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-UCSD birds 200 (2010)
18. Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., Schiele, B.: Latent embeddings for zero-shot classification. In: Proc. of the IEEE conf. on computer vision and pattern recognition. pp. 69–77 (2016)
19. Xian, Y., Lampert, C.H., Schiele, B., Akata, Z.: Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE trans. on pattern analysis and machine intelligence* **41**(9), 2251–2265 (2018)
20. Ye, M., Guo, Y.: Zero-shot classification with discriminative semantic representation learning. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. pp. 7140–7148 (2017)
21. Zhang, Z., Saligrama, V.: Zero-shot learning via semantic similarity embedding. In: Proceedings of the IEEE international conference on computer vision. pp. 4166–4174 (2015)