Zero-Shot Image Classification Guided by Natural Language Descriptions of Classes: A Meta-Learning Approach

R. Lily Hu
Salesforce Research
lhu@salesforce.com

Caiming Xiong
Salesforce Research
cxiong@salesforce.com

Richard Socher
Salesforce Research
rsocher@salesforce.com

Abstract

We propose a model that learns to perform zero-shot image classification from natural language descriptions by using a meta-learner that is trained to produce a correction to the output of a previously trained learner. The model consists of two modules: a task module that supplies an initial prediction, and a correction module that updates the initial prediction. The task module is the learner and the correction module is the meta-learner. The correction module is trained in an episodic approach whereby many different task modules are trained on various subsets of the total training data, with the rest being used as unseen data for the correction module. The correction module takes as input a representation of the task module's training data so that the predicted correction is a function of the task module's training data. The correction module is trained to update the task module's prediction to be closer to the target value. This approach leads to state-of-the-art performance for zero-shot classification on natural language class descriptions on the CUB and NAB datasets.

1 Introduction

The ability to solve a task without receiving training examples – zero-shot learning – is desirable. This ability is particularly valuable when training data is limited, difficult to obtain, and/or expensive. We as humans can learn new tasks from descriptions of the tasks, as we learn from reading encyclopedia entries, manuals, handbooks, textbooks, etc. Our artificial agents should be able to do the same. For language guided zero-shot image classification, this requires grounding language into vision. However, generalizing to novel tasks unseen during training is challenging. The agent must integrate its previous training experience to solve the new task, without individual training samples from the new task, while avoiding over-fitting on its previous training experience.

We propose a model that learns a correction on predictions in the zero-shot setting, based on the training data set used to generate the initial prediction. Hence, our model is called Correction Networks. The meta-learner corrects for the zero-shot predictions of the learner. The intuition for our model is that a zero-shot query sample that is different from samples in the training data will require a different correction than a zero-shot query sample that is similar to samples in the training data.

Correction Networks update the predictions based on the training data. This updated prediction is trained to be closer to the target value than the original prediction. Correction Networks consist of two modules: a *task module* that supplies an initial prediction, and a *correction module* that provides a correction to the initial prediction. The task module is the learner and the correction module is the meta-learner. The final prediction is the task module's initial prediction combined with the correction module's correction. This method is illustrated in Figure 1. The prediction of the meta-learner is used to modify the output of the learner, unlike other meta-learning algorithms that learn to update,

Preprint. Work in progress.

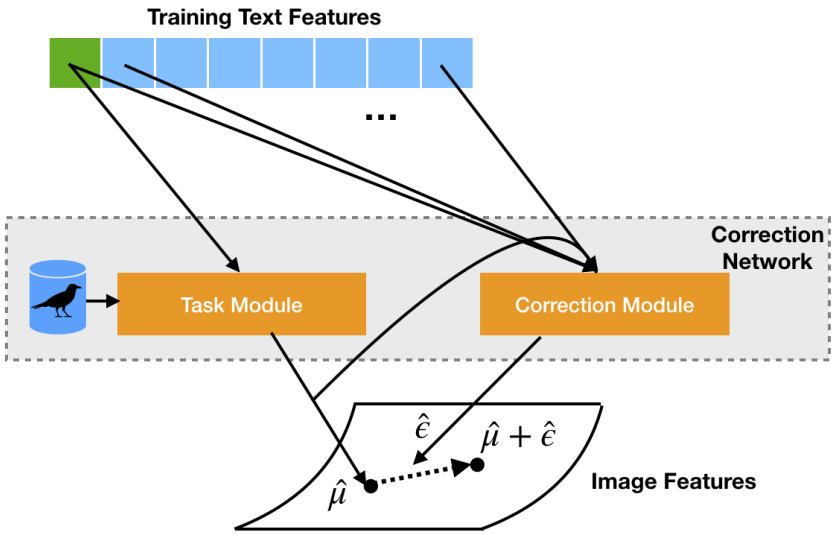

Figure 1: The task module produces an initial prediction. The correction module provides a correction such that when combined with the initial prediction, produces a better prediction.

initialize, or optimize the learner's weights (1) (2). Our approach is for zero-shot learning while previous meta-learning approaches focus on few shot learning and require a few samples. Another novelty is that the correction module, the meta-learner, takes as input the dataset used to train the task module.

Correction Networks can be used for problems for which an updated prediction can be obtained by applying a correction. Correction Networks are independent of the representation of the task module. Existing models that provide predictions can be treated as task modules. The correction module only need the inputs to the task module, the predictions of the task module, and the training data of the task module. While our focus is on neural network architectures, both the task module and correction modules can be, but needs not be, neural networks.

Training proceeds by randomly partitioning the training data D to create disjoint sets D_S and D_U. In the case of zero-shot classification, the partition is done by class so that the classes in D_S and D_U are disjoint. The task module is trained on D_S. After the task module is trained, D_U is used as the 'zero-shot' queries for the task module. The task module supplies initial predictions on D_U. The correction module is trained to predict the corrections that need to be applied to the task module's initial predictions to obtain the true values associated with D_U. One novelty is that the correction module, the meta-learner, also takes as input the dataset used to train the task module.

The main contribution of this paper is a zero-shot learning model that corrects zero-shot predictions based on training data used to generate the initial prediction. We demonstrate our model on zero-shot image classification by grounding natural language descriptions of classes. Our evaluation shows that Correction Networks compares favorably to state-of-the-art zero-shot learning methods.

2 Related Works

For zero-shot learning for images, the majority of state-of-the-art methods are embedding-based methods. This often involves learning a mapping from the visual space to the semantic space of class labels or vice versa. Alternatively, the embedding function between the visual and semantic spaces is jointly learned through a latent space.

To represent the new, novel classes, hand crafted attributes are often used. The idea is that given a defined attribute ontology, each class name can be converted to an attribute vector against which image features are compared (3) (4) (5) (6). Attribute based embedding spaces more recently gave way to text based embeddings. These text-based embeddings learn from a large external text corpus

e.g. Wikipedia or WordNet on a natural language task to learn word embeddings into which class names are then projected (7) (8) (9) (10).

Semantic representations for zero-shot classes have also been created from text documents of the classes eg. Wikipedia articles corresponding to the class. This includes learning attribute representations from a web source by ranking the visualness of attribute candidates (11). Internet sources have also been mined for semantic relatedness of attribute-class association (12). Directly measuring the compatibility between documents describing the class and visual features have also been modeled using deep learning (13) (14) and non-deep learning models (15) (16) (17) (18) (19).

A more recent strategy frames zero-shot recognition as a conventional supervised classification problem by hallucinating samples for unseen classes. Classification performance thus depends on the quality of the hallucinated samples. (20) assumes the visual features of each class is distributed as a Gaussian and estimates the distribution of unseen classes by linearly combining the distributions of seen classes. (21) synthesizes visual data from attributes of classes in a one-to-one mapping. (22) assigns pseudo labels to samples from seen classes. (13) uses generative adversarial training with the generator learning to generate image features from text and query samples are classified based on k-nearest neighbors to labeled or generated samples. In contrast, our approach avoids the data generation phase and instead, directly predicts the center of classes in image feature space. Classification regions are thus defined based on the class of the nearest class center.

Our work is similar to gradient boosting which produces a prediction model using an ensemble of weak models, where each additional model adds an estimator to provide a better model. Unlike gradient boosting, our approach takes in the training data used to create the initial estimate so that the model learns how the initial estimate deviates due to the training data. While the initial learner in gradient boosting is a weak learner, we use a strong learner.

3 Correction Networks

Let D_S denote our training data and D_U our testing data. D_S is subdivided into disjoint sets D_S^s and D_S^u. For classification, the classes in D_S^s are disjoint from the classes in D_S^u. Correction Networks M consists of two modules: a task module M_T and a correction module M_C. The model components and their inputs and outputs are illustrated in Figure 2.

The task module M_T is so-called because it is task specific and related to the application. The task module is trained on D_S^s. The output of M_T is an estimate $\hat{}$ of a target . The predictions of the task module on its training data D_S^s is $\hat{}_S^s$. Training the task module proceeds by minimizing the distance between $\hat{}_S^s$ and the ground truth $_S^s$. A sparsity regularization term is added to the input layer of M_T. The loss is:

$$L_{M_T} = \mathbb{E}[d(M_T(T_S^s);\ _S^s)] + \quad jjw_{M_T;text}\quad jj^2 \tag{1}$$

where T is the class text description, is the empirical mean of samples that belong to the class, and d is a distance function. We use the L2 norm as the distance function.

The task module M_T is not trained on D_S^u nor D_U. The task module's predictions on D_S^u are $\hat{}_S^u$. Likewise, the task module's predictions on D_U are $\hat{}_U$. The correction module M_C computes a correction $\hat{}_S^u$ that is applied to the prediction $\hat{}_S^u$ of the task module M_T, where $\hat{}_S^u$ is calculated based on the data used to train M_T, such that the corrected prediction $(\hat{}_S^u + \hat{}_S^u)$ is closer than $\hat{}_S^u$ to the ground truth $_S^u$. Training the correction module proceeds by minimizing the distance between $_S^u$ and $(\hat{}_S^u + \hat{}_S^u)$. We use the L2 norm. The training data for the task module D_S^s is input into the correction module by representing the training data D_S^s as an un-ordered collection of data by using a pooling function. We use linear layers followed by sum pooling. The objective function of the correction module is to minimize:

$$L_{M_C} = \mathbb{E}[d(M_C(T_S^u);\ _S^u \quad M_T(T_S^u))] \tag{2}$$

We adopt the meta-learning sampling strategy for training as in (23). Training data for Correction Networks are formed by randomly selecting a subset D_S^s from the training data D_S. Then, the task module M_T, is trained on D_S^s. The remaining tasks that the task module M_T does not train on are treated as D_S^u for M_T. The algorithm is detailed in Algorithm 1 in Appendix B.

To use Correction Networks for evaluation, the task module M_T outputs $\hat{}_U$ and the correction network supplies $\hat{}_U$. The output of the Correction Networks is $_U = \hat{}_U + \hat{}_U$. The evaluation

Table 1: Zero-shot learning classification results accuracy @ 1 on the CUB-200-2011 dataset and the NAB dataset using class descriptions from Wikipedia on the Super-Category-Shared (SCS) and Super-Category-Exclusive (SCE) zero-shot splits

METHOD	CUB		NAB	
	SCS	SCE	SCS	SCE
MCZSL (26)	34.7	-	-	-
WAC-Linear (15)	27.0	5.0	-	-
WAC-Kernel (19)	33.5	7.7	11.4	6.0
ESZSL (16)	28.5	7.4	24.3	6.3
SJE (27)	29.9	-	-	-
ZSLNS (18)	29.1	7.3	24.5	6.8
$\text{SynC}_{\text{fast}}$ (28)	28.0	8.6	18.4	3.8
SynC_{OVO} (28)	12.5	5.9	-	-
ZSLPP (29)	37.2	9.7	30.3	8.1
GAZSL (13)	43.7	10.3	35.6	8.6
Correction Networks	**45.8**	10.0	**37.0**	**9.5**

algorithm is outlined in Algorithm 2 of Appendix B. Correction Networks can be used for zero-shot problems for which an updated prediction can be obtained by applying a correction. Appropriate problems are those with continuous valued outputs. For classification, samples can be represented as continuous values in a feature space and then classified based on the nearest class center. In our experiments, both the base network and the Correction Network are deep neural networks.

4 Experiments

4.1 Task Setup

We demonstrate Correction Networks on fine-grained zero-shot classification based on natural language text descriptions of the class. We evaluate our method on Caltech UCSD Birds 2011 (CUB) (24) and North America Birds (NAB) (25) with class data from Wikipedia. Additional details of the data and experiments are found in Appendix C.

4.2 Conventional Zero-Shot Recognition

The top-1 accuracy of our method and eight state-of-the-art algorithms for the CUB and NAB datasets for both the SCS split and the SCE split are tabulated in Table 1. The eight comparison models are MCZSL (26), ZSLNS (18), SJE (27), WAC (19), SynC (28), ZSLPP (29), and GAZSL (13). The performance numbers are copied from (13). MCZSL directly uses manual part annotations to extract visual representations. On the other hand, our approach, GAZSL, and ZSLPP merely use detected parts for both training and testing. Thus, methods that use detected parts are expected to perform poorer than MCZSL, which uses manual parts annotations. Our model performs favorably against the other models, showing a relative improvement of up to 10% over the previous state-of-the-art. Qualitative results are shown in Figure 4 in Appendix E. Ablation studies are described in Appendix D.

4.3 Generalized Zero-Shot Learning

The conventional zero-shot learning setting considers queries that come from unseen classes D_U and classification of queries is restricted to be among the unseen classes U. In contrast, the generalized zero-shot learning setting classifies queries from both seen D_S and unseen classes D_U into $S \ [\ U$. A metric for generalized zero-shot learning performance is the area under the seen-unseen curve (30). This is tabulated in Table 2. For Correction Networks, Correction Networks is used to predict ^ for the unseen classes while the empirical is used for the seen classes. Correction Networks consistently outperforms previous approaches in the generalized zero-shot learning setting. This improvement is between 3-31% relative to the runner-up.

Table 2: Generalized Zero-shot learning classi cation area under Seen-Unseen Curve on CUB and NAB datasets

METHOD	CUB		NAB	
	SCS	SCE	SCS	SCE
WAC-Linear (15)	23.9	4.9	23.5	-
WAC-Kernel (19)	22.5	5.4	0.7	2.3
$SynC_{Fast}$ (28)	13.1	4.0	2.7	0.8
ESZSL (16)	18.5	4.5	9.2	2.9
ZSLNS (18)	14.7	4.4	9.3	2.3
$SynC_{OvO}$ (28)	1.7	1.0	0.1	-
ZSLPP (29)	30.4	6.1	12.6	3.5
GAZSL (13)	35.4	8.7	20.4	5.8
CorrectionNet	41.9	9.0	25.4	7.6

5 Conclusion

We propose a zero-shot learning model that consists of a task module and a correction module that is trained to extrapolate from training data to unseen data. This model is demonstrated on image classi cation guided by natural language descriptions of novel image classes. The training data is partitioned into a set of data used to train the task module and a disjoint set of data used to train the correction module. This later data is zero-shot with respect to the task module and the correction module is trained to update the task module's zero-shot predictions to create a better prediction.

Our model performs favorably against the state-of-the-art on zero-shot classi cation. Additional investigations include applying Correction Networks to other tasks with outputs that can be updated, including for example, estimates for regression problems, classi cation probabilities, or probability distributions for reinforcement learning policies. This framework is exible to different representations of the task and correction modules and can be extended to other types of model predictions that can be updated to improve predictions.

Acknowledgments

The authors would like to thank Nitish Keskar.

References

[1] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 1126–1135, 2017.

[2] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *International Conference on Learning Representations*, 2017.

[3] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 951–958, IEEE, 2009.

[4] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," in *Advances in neural information processing systems*, pp. 1410–1418, 2009.

[5] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1778–1785, IEEE, 2009.

[6] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for attribute-based classi cation," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 819–826, IEEE, 2013.

[7] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pp. 1717–1724, IEEE, 2014.

[8] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng, "Zero-shot learning through cross-modal transfer," in Advances in neural information processing systems, pp. 935–943, 2013.

[9] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al., "Devise: A deep visual-semantic embedding model," in Advances in neural information processing systems, pp. 2121–2129, 2013.

[10] Y. Fu, T. M. Hospedales, T. Xiang, Z. Fu, and S. Gong, "Transductive multi-view embedding for zero-shot recognition and annotation," in European Conference on Computer Vision, pp. 584–599, Springer, 2014.

[11] T. L. Berg, A. C. Berg, and J. Shih, "Automatic attribute discovery and characterization from noisy web data," in European Conference on Computer Vision, pp. 663–676, Springer, 2010.

[12] M. Rohrbach, S. Ebert, and B. Schiele, "Transfer learning in a transductive setting," in Advances in neural information processing systems, pp. 46–54, 2013.

[13] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal, "A generative adversarial approach for zero-shot learning from noisy texts," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[14] L. J. Ba, K. Swersky, S. Fidler, and R. Salakhutdinov, "Predicting deep zero-shot convolutional neural networks using textual descriptions.," in ICCV, pp. 4247–4255, 2015.

[15] M. Elhoseiny, B. Saleh, and A. Elgammal, "Write a classi er: Zero-shot learning using purely textual descriptions," in Computer Vision (ICCV), 2013 IEEE International Conference on, pp. 2584–2591, IEEE, 2013.

[16] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in International Conference on Machine Learning, pp. 2152–2161, 2015.

[17] Z. Fu, T. Xiang, E. Kodirov, and S. Gong, "Zero-shot object recognition by semantic manifold distance," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2635–2644, 2015.

[18] R. Qiao, L. Liu, C. Shen, and A. van den Hengel, "Less is more: zero-shot learning from online textual documents with noise suppression," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2249–2257, 2016.

[19] M. Elhoseiny, A. Elgammal, and B. Saleh, "Write a classi er: Predicting visual classi ers from unstructured text," IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 12, pp. 2539–2553, 2017.

[20] Y. Guo, G. Ding, J. Han, and Y. Gao, "Synthesizing samples fro zero-shot learning," IJCAI, 2017.

[21] Y. Long, L. Liu, L. Shao, F. Shen, G. Ding, and J. Han, "From zero-shot learning to conventional supervised classi cation: Unseen visual data synthesis," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[22] Y. Guo, G. Ding, J. Han, and Y. Gao, "Zero-shot learning with transferred samples," IEEE Transactions on Image Processing, vol. 26, no. 7, pp. 3277–3290, 2017.

[23] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in Advances in Neural Information Processing Systems, pp. 4077–4087, 2017.

[24] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Tech. Rep. CNS-TR-2011-001, California Institute of Technology, 2011.

[25] G. Van Horn, S. Branson, R. Farrell, S. Haber, J. Barry, P. Ipeirotis, P. Perona, and S. Belongie, "Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 595–604, 2015.

[26] Z. Akata, M. Malinowski, M. Fritz, and B. Schiele, "Multi-cue zero-shot learning with strong supervision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 59–68, 2016.

[27] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele, "Evaluation of output embeddings for fine-grained image classification," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pp. 2927–2936, IEEE, 2015.

[28] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha, "Synthesized classifiers for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5327–5336, 2016.

[29] H. Z. Mohamed Elhoseiny, Yizhe Zhu and A. Elgammal, "Link the head to the" beak": Zero shot learning from noisy text description at part precision," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[30] W.-L. Chao, S. Changpinyo, B. Gong, and F. Sha, "An empirical study and analysis of generalized zero-shot learning for object recognition in the wild," *European Conference on Computer Vision*, pp. 52–68, Springer, 2016.

Appendix A Modules with inputs and outputs

Figure 2: Correction Networks consists of two modules: a task module and a correction module, demonstrated on zero-shot image classi cation based on a natural language description of the zero-shot class. The task module takes as input the natural language description of the zero-shot class and makes an initial prediction of the cluster center of the class in image feature space. The correction module improves this initial prediction by applying a correction, taking as input the task module's initial prediction, training data, and the zero-shot class description.

Appendix B Algorithm

Algorithm 1 Correction Networks training algorithm. K is the number of classes in the training set. S is partitioned into S^s and S^u by class. T_S denotes the class descriptions of classes in S. D_S is the set of individual samples of x_i (y_i) for the classes in the training set. D_S^s denotes the subset of D_S containing all samples from classes in S^s. D_k is the subset of D_S containing the individual elements $(x_i\ y_i)$ such that $y_i = k$. MEAN is a function that returns the mean of it's inputs. OPTIMIZER is an optimizer. d is a distance function, for example, L2 distance. M_T is the task module. M_C is the correction module.

for k to K do
  $\mu_k \leftarrow$ MEAN(D_k)
end for
while not done do
  $S^s \leftarrow$ RANDOMSAMPLE$(\{1, ..., K\}, s)$
  $S^u \leftarrow \{1, ..., K\} \setminus S^s$
  while M_T has not finished training do
    $\hat{\mu}_S^s \leftarrow M_T(T_S^s)$
    $J_{M_T} \leftarrow d(\hat{\mu}_S^s, \mu_S^s) + \lambda \|w_{M_T}\|^2$
    $w_{M_T} \leftarrow$ OPTIMIZER$(\nabla_{w_{M_T}} J_{M_T}, w_{M_T})$
  end while
  $\hat{\mu}_S^u \leftarrow M_T(T_S^u)$
  $\hat{\mu}_S^u \leftarrow M_C(T_S^u, \hat{\mu}_S^u, T_S)$
  $\mu_S^u \leftarrow \mu_S^u - \hat{\mu}_S^u$
  $J_{M_C} \leftarrow d(\hat{\mu}_S^u, \mu_S^u)$
  $w_{M_C} \leftarrow$ OPTIMIZER$(\nabla_{w_{M_C}} J_{M_C}, w_{M_C})$
end while

Algorithm 2 Correction Networks evaluation algorithm. S is the set of classes in the training set. U is the set of zero-shot classes. T_U denotes the class descriptions of zero-shot classes in U.

Require Trained $M_T(T_S)$
Output: Prediction μ_U

$\hat{\mu}_U \leftarrow M_T(T_U)$
$\hat{\mu} \leftarrow M_C(T_U, \hat{\mu}_U, T_S)$
$\mu_U = \hat{\mu}_U + \hat{\mu}_U$

Appendix C Datasets

Image Data We evaluate our method on Caltech UCSD Birds 2011 (CUB) (28) and North America Birds (NAB) (25). Two splits were proposed for each dataset (29). The splits are named Super-Category-Shared (SCS) and Super-Category-Exclusive (SCE). The SCE splits are more difficult than the SCS splits. The visual features are activations extracted from the FC layer of a pre-trained parts detector (29) and are the same as used in (13).

Text Data Each class is associated with a Wikipedia article. The articles are tokenized into words, stop words are removed, and words are reduced to their word stems. Then, the processed text is represented by TF-IDF features. We use the same text features as (13). Examples of the text and images are shown in Figure 3.

Figure 3: Examples from the dataset where there is one natural language description of a class and the goal is to classify images for new classes without training sample images.

Appendix D Ablation Studies

To examine the contributions of separate components of our model, we conduct ablation studies by removing selected components. Then, the model is retrained and evaluated on the test set. The resulting performance of the ablated models are reported in Table 3. Removing the correction module degrades the performance but the task module itself already achieves the state of the art. The correction module improves the performance by 2.0%. This suggests that the correction modules makes a slight change to the output of the task module. When we remove the task module's training data as input into the correction module, the zero-shot accuracy decreases by 2.4%. This demonstrates that the task module's training data is important to the correction module's prediction.

Table 3: Effects of different components on zero-shot classification accuracy (%) on CUB SCS

METHOD	CUB
Task module only	43.8
Correction module without task module's training dataset	43.4
Correction Networks	45.8